Energy-aware encryption mechanism for m-commerce devices

Hamad, Fadi

Award date:
2009

Awarding institution:
Coventry University

ENERGY-AWARE ENCRYPTION MECHANISM FOR -COMMERCE DEVICES

FADI MOHAMMAD HAMAD

A thesis submitted in partial fulfilment

of the University's requirements

for the Degree of Doctor of Philosophy

SEPTEMBER 2009

Coventry University

Abstract

With the wide spread of mobile phones, PDAs, and Smartphones, M-Commerce has become a major application domain for mobile devices, unlike conventional wired networks, mobile devices allow the user to conduct online transactions regardless of the time and the place as long as there is mobile network coverage. However, online transactions require adequate level of security to insure the confidentiality, the integrity, and the availability of the user's information. Security measures consume a considerable amount of energy and require more time in processing.

The aim of this thesis is to optimise the energy and the resources consumption of mobile phones when applying variant symmetric and asymmetric schemes. This aim can be achieved through developing A System State Security Management Framework, SSSM, which will implement encryption schemes, symmetric and asymmetric, and will provide different options to enable the user to choose the type of encryption, the key size, and number of rounds of computation to optimise the energy consumption level of the mobile phone. This thesis compares the power and the resources consumed by the most commonly used encryption algorithms such as CAST, IDEA, Triple-DES, RSA, and AlGamal. This comparison helps to draw the advantages and disadvantages of each algorithm scheme used in reference to the security level it provides and the power it consumes.

Implementing this mechanism will enhance the performance of mobile phones by increasing the security levels provided by the encryption schemes and utilising the limited power and resources efficiency. Therefore, confidentiality will be presented in mobile phones and variant encryption schemes, symmetric and asymmetric, and changeable key sizes and rounds, will ensure the authenticity of both senders and recipients depending on their needs as well as resources available. This research makes contributions in two major areas; the first area consists of the novel Energy Aware Encryption polices generated by this work, the second area of contribution is the energy measurements and experimental results which validate the approach presented in the research.

Declaration

This thesis is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

Acknowledgement

My deepest appreciations undoubtedly go to my supervisory team Dr Leonid Smalov, and Professor Anne James, to whom I am grateful for their help, guidance, and encouragement throughout this work. Many thanks also to Dr Mohammad Hammoudeh for his support in the final stages of writing this thesis.

Special thanks go to Sarah Fogarty and Ann Daly for helping me and providing with a valuable academic research inquiries.

Further thanks are due to my parents for their unlimited love and support. They taught me the value of knowledge and have stood by me at all times, providing constant support, encouragement and love.

Finally, I would like to thank my wonderful Siblings, Lara, Bashar, and Harb, for their patience and support, their understanding and encouragement gave me the momentum to complete my PhD study.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1 Introduction

1.1 Focus and Approach of the Research

The emergence of wireless and mobile networks has made possible the introduction of electronic commerce to a new application and research area: mobile commerce, M-commerce. Internet-enabled mobile handheld devices are one of the core components of a mobile commerce system, making it possible for mobile users to directly interact with mobile commerce applications. (Hu et al, 2005)

Security is provided through security services (Khosrow-Pour et al, 2006). Confidentiality of Information prevents unauthorised users from reading the data and allows only authorised users to do so. Confidentiality is achieved through encrypting the transmitted data in wireless communication and maintaining the encryptions keys used. Authentication is another necessary measure which ensures the correct identification of the source message. As for the receiving end, Integrity ensures the data has not been tampered by unauthorised parties during transmission. The information availability is restricted to allowed parties by Access Control; The communication must be available to both the sending and the receiving parties when needed, while non-repudiation ensures that neither the origination nor the receptor of the information can deny the transaction.

The growth of wireless networks restricted by the security level required by every party (Gutierrez, 2009). These requirements are determined by the organisation using the wireless network; financial companies require very strong security measures to ensure that their information are kept safe from unauthorized parties to maintain information confidentiality. On the other hand, Internet access points such as hot-spots networks only require legitimate users access the network without the need to ensure neither confidentiality nor data integrity, and in the case of public internet access networks, no security is required.

Conventional or wired networks' security has come a long way in providing Confidentiality, Integrity, Authentication, and non-repudiation. Wired systems are inherently more secure than the wireless systems because of the wired connectivity (Siemens, 2008). Threats such as eavesdropping, and port scanning, require the malicious intruder to be physically connected to the network, either through an insider, or an outsider connected through dial up, VPN, ...etc.

However, access to wireless network, does not require a physical connection as the world the connection to a LAN is not limited by the requirement of physical connectivity; the nature of wireless networks increases the possibility of intruders to eavesdrop and conduct a malicious attack to such networks. Intruders can easily listen to transmitted packets through wireless interfaces, and allow them to inject malicious packets which might disrupt the network and can lead to the possibility of compromising the confidentiality and the integrity of information without even being inside the premises. Wireless Networks are prone to the same vulnerabilities affecting the wired networks, but the nature of the wireless networks compromises the security measures of those of the wired networks. In a study conducted by BT in 2005, they concluded that unprotected wireless networks' security is compromised in seconds, and wireless networks protected by wired equivalent privacy (WEP) can still be hacked in a matter of minutes, (BT, 2005). In addition, human's error and misfortune compromises all security measures if the mobile device is lost or misplaced and found by malicious parties. Hence, the security mechanisms implemented for the wired systems cannot be used directly in the wireless environment.

Security measures are provided through security protocols implemented at the transmission level, such as frequency hopping, and spread spectrum technologies, are considered very expensive to implement by users and organisations (Yahya et al, 2006), in addition to the fact that Linear Feedback Shift Registers (LFSRs) (Kalligeros et al, 2002) are easy to break due to mathematical analyses developed in the past to analyse LFSR (Lablans, 2005). As a result, developers have drawn their attention to developing data encryption protocols such as the WPA protocol for security (Wong, 2003) which operates at the data link layer, IP Security (IPSec) which operates at the network layer (InterLink Networks, 2003), and Secure Socket Layer (SSL) operated at the transport layer to secure transactions on the Internet (Entrust, 2007). All these protocols rely on encryption or encryption related mechanisms to provide the security services. Encryption in this sense is thus the backbone of security services.

The protocols mentioned above sound familiar, because they were developed to protect wired networks, but their implementation on wireless networks have raised quite few concerns in reference to the limited battery power, memory, and processing capabilities of mobile devices. Thus, the need to study the encryption schemes and design an energy aware encryption system to tailor the needs of wireless environment has become necessary.

1.2 Problem Context

In general, Mobile handheld devices' features, such as mobile access, rapid network configuration, and lack of wires, make wireless networks particularly attractive (Karygiannis and Owens, 2002). The availability of mobile technology has enabled users to store their private data such as personal information, emails, bank account details, addresses ...etc. this has provided malicious parties a great incentive to consolidate their efforts to intercept wireless communication and obtain the data exchanged through these networks (Narula et al, 2007). Wi-Fi technology is a broadcast radio technology that works on the same 2.4 gigahertz microwave radio band as modern cordless telephones and Bluetooth wireless devices built into many notebook computers. As a result, Wi-Fi has the same advantages and disadvantages as any radio technology (Kashi, 2004).

On the other hand, Mobile devices have lots of components that drain the battery; there is energy needed to operate the diversity of available radios like ultra wide band (UWB), Radio Frequency Identification (RFID), FM radio, 2G/3G cellular radio transmitter and receivers, wireless LAN (WLAN), Bluetooth, Digital Video Broadcasting for Handhelds (DVBH) and GPS. According to (Chang, 2008) most of the power is consumed when the phone is used for making voice calls. The radio frequency (RF) part that transmits the voice to the base station uses power amplifier (PA) (Chang, 2008). PA consumes most of the power. Research is going on in this RF area for finding out the solution to reduce power. PA gets more attention from mobile companies worldwide. Apart from the cellular part, radio components like Bluetooth, WLAN, etc also consumes more power. For example, while using WLAN, the packet loss is about 30% which leads to lot of power consumption (Chang, 2008). From the use cases perspective, the other major energy consuming items of the mobile devices are the display, mass memory and application processor. In the future, the power consumption growth will be in local connectivity and imaging elements as the ad-hoc networks are expected to grow and rich multimedia applications demand high quality imaging. It is crucial to ensure that security measures implemented on mobile devices have a minimum effect on both computational efficiency and battery lifetime.

Encryption forms the basic building block for various security services (Delfs, 2007). It is thus vital to analyse the energy consumption levels for encryption schemes and considering the various options which inversely affect the energy drainage like key sizes, number of rounds, layers of security, and the amount of data processed per packet. Knowledge of the tradeoffs would also help in designing systems that can adapt the security of the communication link based on the device being used and the battery power left on it. The harsh wireless environment

further complicates trade-off in terms of power and security. There has not been any research that studies the tradeoffs between security of wireless devices and the battery power consumed by various encryptions of the algorithms.

From the above scenario, firstly, we address the importance of encryption implementation to secure information against malicious individuals. Secondly we identify that the computational capabilities and battery lifetime limitations of mobile devices are decisive factors with regard to security strategy, as security measures require too many calculations which consume a considerable amount of energy. Thirdly, we recommend an energy awareness strategy to provide an adequate level of security to the mobile user with reference to low power and resources consumption. This thesis focuses on providing solutions to achieve adequate security levels for data exchange while minimising the energy expenditure within a mobile communication environment with requirements for low consumption of power and resources.

1.3 Research Hypothesis

Let us examine the main problems that we face with achieving an adequate security level of information and data exchange over wireless and mobile networks. If we consider traditional exchanges of goods or money transactions, security in such cases is achieved quite trivially. For instance, when shopping in a supermarket, a customer takes away the purchased goods at the time he makes a payment at the counter. When signing a contract, the parties involved sign the document at the same place and time, or use the solicitors as trusted third parties to make sure the contract is signed and exchanged fairly. In all these cases, fairness is achieved by exchanging the items simultaneously.

There have been studies that compare the performance of some of the encryption and decryption schemes in terms of bytes processed per unit time or time for operation (Tamimi 2005, Mancillas-López1 2007). However, there have not been enough studies related to the energy consumed by the encryption schemes in every day communication environment on mobile devices.

Having analysed the transaction made by the mobile device user, we have noticed a common feature, when an exchange of encrypted data or information, the user relies on the mobile resources valid at the time of exchange. From the above, the research hypothesis is examining

the efficiency of information and data encryption transmitted through mobile devices in relation to power consumption; in other words, is a secure exchange of information could be achieved by encrypting the information or data in a specific way in reference to battery, memory, and storage status and according to sender's needs?

1.4 Aim and Objectives

The aim of this thesis is to overcome the lack of security when exchanging valuable information and data over the wireless and mobile environment and to develop an energy-aware security policy to maintain adequate security levels and non-repudiation services required for safeguarding M-commerce transactions while ensuring a low level of energy consumption.

This thesis advocates a framework for providing secure energy-aware data communication while sustaining low energy expenditure. A System State Security Management Framework, SSSM, adapts the security level depending on the amount of available energy levels, and the amount of processing time needed to execute the security policy. By ensuring these two services, we aim at helping business parties overcome the distrust and increasing the confidence while engaging in ad-hoc M-commerce transactions with other parties over the wireless and mobile networks, and increase their awareness to security and energy saving during a transaction when using mobile handheld devices.

The objectives of this thesis are:

(1) Review power consumption and management in devices used for communication in M-Commerce

(2) Analyse commonly used encryption algorithms in M-Commerce

(3) Assess the effect on power consumption of various commonly used encryption algorithms in M-Commerce

(4) Assess the security levels afforded by various commonly used encryption algorithms in M-Commerce through the following suggested scenarios:

– **High security level**: an exchange of highly important and confidential information or data by two parties, which can affect business ventures, together with its non-repudiable proof, where the aim is to generate a proof of reception of a specific text and no content assurance is needed nor can be guaranteed.

This scenario requires asymmetric encryption scheme due to the sensitive information exchanged between two business men, this information can include bank accounts, addresses, personal information, classified information ...etc

– **Intermediate security level**: an exchange of important information and data, but no direct effect on business ventures. This service is not required for non-repudiable proof.

Symmetric encryption schemes are required for this scenario, as the information exchanged between the users are important but does not include business accounts, personal information, or any kind of classified information.

– **No security level Scenario**: General information and data are exchanged which are not potentially endangering mobile users if exposed.

This scenario represents day to day information exchange between users where no important information is exchanged; therefore, no encryption scheme is required.

(5) Develop an energy-aware security system to be used in M-Commerce that recommends appropriate security levels for various transactions and which allows the user to make the final decision on security level.

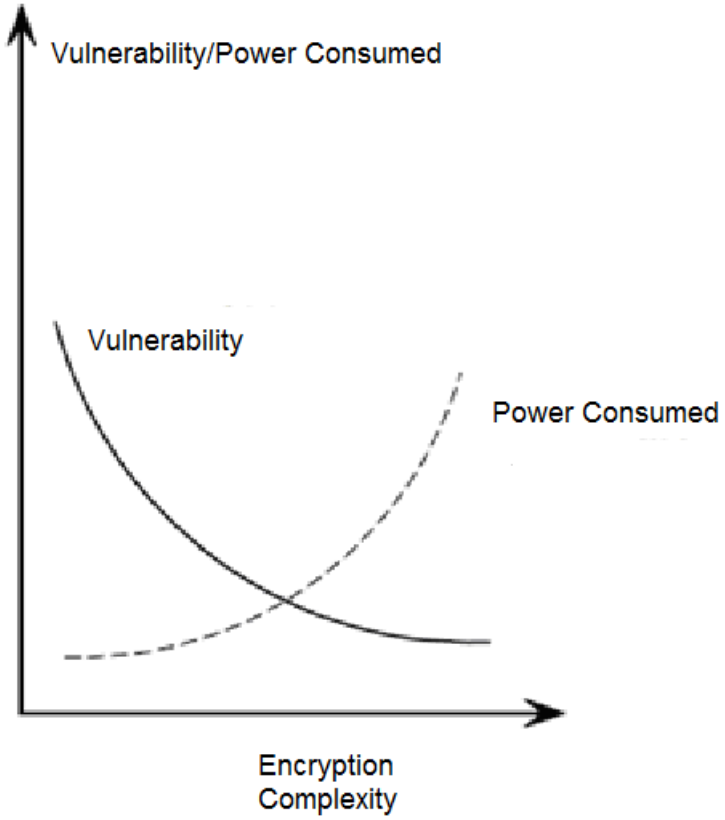

Figure 1: Encryption Vs Power Consumption

The primary challenge in providing security in low power mobile wireless devices lies in the conflicting interest between minimizing power consumption and maximizing security. In general, we can safely assume that by doing more computations one can achieve higher amount of security. For example, the strength of encryption schemes depend on complexity of the encryption algorithm, more complex algorithms produces higher levels of security at the cost of additional power consumption. For example Figure 1 illustrates the relationship between vulnerability and the power consumption in reference to the encryption complexity. Therefore, in order to design power efficient encryption algorithms for mobile handheld devices there is an inherent need to understand the relationships between power consumption and encryption parameters. Once these relationships are understood well then it is possible to optimize power consumption with reference to a security requirement or vice-versa.

1.5 Methodology Adopted

The base of submission to PhD is framework which provides application and Energy Aware mechanism with dynamic balancing of encryption schemes and energy reservation for mobile devices such as mobile phones. This section describes the methodology adopted in conducting and evaluating this research. This includes studying and reviewing related research, analysing encryptions, designing and executing an experiment which will inform the design of the SSSM.

In order to meet the aim of developing an energy aware security system for use in M-Commerce the following general method was used:

1. Evaluation on power consumption and management in mobile devices through literature review.

2. Analysis of relevant encryption algorithms through literature review and investigation.

3. Identification of M-Commerce scenarios which require variant levels of security.

4. Design of experiments to compare power consumption of various encryption algorithms on mobile devices.

5. Execution of Experiments.

6. Evaluation of results.

7. Design of energy-aware security system based on experimental results.

8. Evaluation of energy-aware security system.

The experimental and development part of this research involves:

- Running the variant encryption algorithms on a laptop environment to obtain a reference point for Symbian S80 series Smartphones and PDAs.

- Comparing the results to those obtained from the laptop experiments, and creating reference points for evaluating the energy awareness strategy when implemented; running the variant encryptions by following the same order used on the laptop environment to measure the difference of energy efficiency performed on both the Smartphone and PDA environments. This allows us to draw a conclusion on how efficient the encryptions are in the presence of a limited source of energy.

- Developing driver software, written in C++, to optimize the energy and the resources consumption of Symbian S80, open source operating system widely used for Smartphones, mobiles, and PDAs , when applying variant symmetric and asymmetric schemes.

In order to evaluate the results, three scenarios will be considered in this research .A suitable encryption scheme for each scenario will be implemented on a Laptop, a PDA, and a Smart Phones. The results will reflect the best possible energy saving scheme for each scenario, to be adopted on each devices used in the research.

The results obtained in the experimental work inform the design of the SSSM as they provide evidence of efficiency savings obtainable through utilisation of variant encryption schemes depending on application need.

The flow chart for the software code suggested to be used is presented in figure 2. The battery and computational exchange of encryption schemes under different scenarios are considered in various experimental setups but the underlying setup remains the same. Initialization in case of encryption would be to establish the keys required. For the purpose of comparing their performance, a big file of size 5MB unless explicitly mentioned is processed multiple number of times. The flow chart explains the encryption process, where complexity is depending on the battery's percentage. In chapter five, we will discuss in details the mechanism in choosing the most appropriate security measures according to the users' needs.

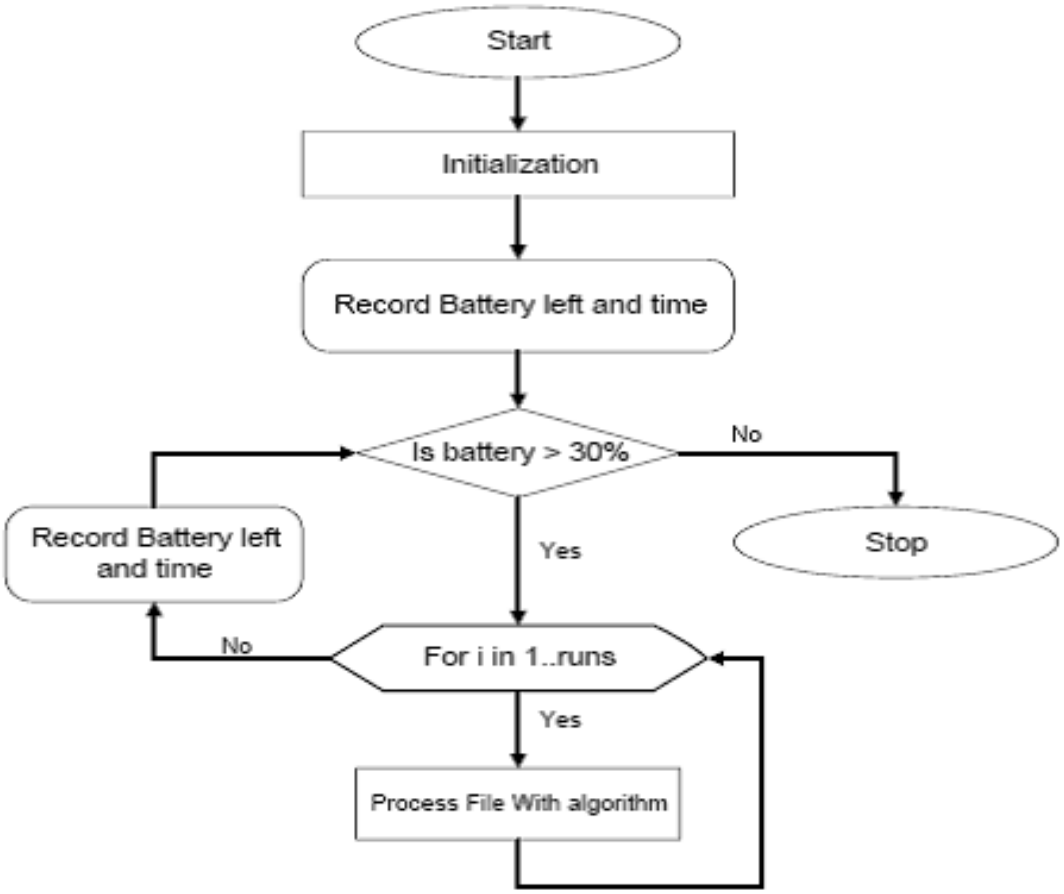

Figure 2: Flow Chart for Driver Program in the Experiments

Processing in experiment for encryption is to read data from the file, then encrypt the data and put it in another file. This process will continue till the battery drains to 30% of the lifetime left. We will stop at 30% because after that the systems alarm and data recovery mechanisms become active and the performance of the schemes change. After a few runs of processing on the file the battery life left and the system time is recorded. The average battery life consumed per run and the time taken to do so is the calculated results.

1.6 Organisation of Thesis

The work in this thesis describes progress towards providing a complete Energy-aware Framework for Mobile Handheld Devices with the capability to secure the transmitted information and data using the various communication means.

Chapter 2: Power Consumption and Management of Mobile Devices.

This chapter builds upon a basic knowledge of power consumption and management of mobile devices. First, Mobile Devices limitations and Mobiles' components which drain the batteries are outlined. Next an overview of battery types in mobile devices and the rate of battery's development are addressed. Then power consumption and battery management are discussed.

Chapter 3: Information Security.

This chapter discusses the network security, Confidentiality, Integrity, and Authentication. It also addresses the non-repudiation aspect of security. Next SSL, IPSEC, and WPA are viewed.

Chapter 4: Encryption Algorithms.

This chapter describes the symmetric encryption, and discusses three different types of symmetric encryptions, the IDEA, CAST, and AES. Next the asymmetric encryption algorithms are discussed with examples such as RSA, ElGamal, and ECIES.

Chapter 5: A System State Security Management Framework.

Chapter 5 describes A System State Security Management Framework, and discusses the various components which affect the battery and resources consumption. And shows the determinants and policies applied

Chapter 6: Methodology.

This chapter shows the method followed and the experiment conducted to obtain the results, then describes three scenarios which are considered in this thesis.

Chapter 7: Laptop Results.

Chapter 7 shows the results obtain from the laptop experiment. A comparison between symmetric encryptions and asymmetric encryptions are drawn in reference to Time and Energy consumed.

Chapter 8: Handheld Devices Results.

In this chapter, a complete comparison is drawn between the algorithms studied in this thesis, and suggestions are made to determine which level of security is recommended to the users.

Chapter 9: Conclusion & Future work

Conclusion is drawn and future work is suggested in this chapter.

Chapter 2

Power Consumption and Management of Mobile Devices

2 Power Consumption and Management of Mobile Devices

2.1 Introduction

Mobile devices have lots of components that drain the battery. There is energy needed to operate the diversity of available radios like ultra wide band (UWB), FM radio, 2G/3G cellular radio transmitter and receivers, Bluetooth, Digital Video Broadcasting for Handhelds (DVBH) and GPS. Most of the power is consumed when the phone is used for making voice calls. The radio frequency (RF) part that transmits the voice to the base station uses power amplifier (PA). PA consumes most of the power. Research is going on in this RF area for finding out the solution to reduce power. PA gets more attention from mobile companies worldwide. Apart from the cellular part, radio components like Bluetooth, WLAN, etc also consumes more power. In this Chapter we explore Mobile devices' limitations, battery technology, and power consumption management.

2.2 Power Consumption

Power consumption is a restraining factor which limits the functionality offered by portable devices, mobile hand held devices, that operate on batteries (Hua et al, 2006). Functionality, high performance processing, battery lifetimes longevity, smaller size, and low prices, are considered leading factors in M-Commerce industry. Unfortunately, the battery technology is not progressing as fast as the digital technology or the increasing user demand. On an average there is 2% increase in battery efficiency every year (Koehler et al, 2005). Portable devices are also getting smaller and smaller, implying that the amount of space for batteries is also decreasing. Decreasing size results in less energy storage and a need for less power consumption. Users do not accept a battery lifetime of less than an hour; for wrist-watch-like devices even lifetimes of several months are expected. New, more powerful processors appear on the market that can deliver the performance users desire for their new applications.

Unfortunately, these powerful processors often have higher power consumption than their predecessors. Finally, users do not only want more processing, but also more features such as multimedia, mass storage, always-on wireless access, and speech recognition. It is important to utilize the available energy within batteries as efficiently as possible to meet user demands. Energy preservation, or energy management, is further translated into low power consumption

by all parts of a portable device. Sophisticated power management is an important requirement to increase the battery life, usability, and functionality of devices.

Power consumption is the rate at which energy is consumed. With the fixed energy capacity of a battery, the power consumption directly determines the battery lifetime of a portable device. There is a difference between power and energy. The ideal situation is to maximize performance while minimizing energy consumption. Efficiency is the key to solving the power crisis. High performance with high power consumption does not necessarily mean less energy efficient and conversely, low performance and low power consumption does not mean that a device is more energy efficient 92. For example, a component may consume five times more power and deliver ten times the performance of alternatives. Such a component would double the efficiency. The primary concern is improving the energy efficiency, even if it requires that we temporarily increase power consumption.

The power consumption of portable devices is not dominated by a single component. Several studies have investigated the power consumption of portable devices (Shearer 2007, Sorber 2004, Pouwelse 2003) The main conclusion is that there is no single component or single activity that dominates the power consumption in a portable device. Therefore, the power consumption of all components needs to be reduced to lower the total amount of power. This section briefly discusses the problems and future technology directions for a number of components of portable devices (Vihmalo, 2005). These components form the basis of wireless multimedia playback devices as shown in Figure 3.

Figure 3. The components of wireless multimedia playback device used in a previous research (Pouwelse, 2003)

2.3 Battery Technology

Batteries used in wireless devices are an essential component. The batteries used with current technology provide a lifetime of 1.5 hours to 4 hours. With people using more and more of the

wireless devices there is a heavy demand for longer lasting batteries of less weight and smaller size. Unfortunately, the battery technology is not progressing as fast as the digital technology or the increasing user demand. On an average there is 2% increase in battery efficiency every year (Hallmark, 2002). In this section we will look at various types of batteries that are being used in today's market.

Various technologies exist today for batteries. Table 1 provides the characteristics of some of the existing battery technologies. Voltage presents the amount of power that can be delivered by the battery and a milli-amperes hour (mAh) is the time for which that power can be delivered. Higher voltage is preferred since if battery has lower output voltage then additional circuit is required to convert the voltage to higher values.
Explain weight energy density and volume energy density

Table 1: Characteristics of Major Battery Systems (Delfs, 2007)

Rechargeable Nickel-Cadmium (NiCd) batteries are popular for use in cordless phones and other moderate power devices. These batteries have a 'memory effect', which means they remember the previous discharge level when recharged and may play dead when they reach the state again. Hence it is required that they should be recharged after fully discharging. Nickel-metal hydride (NiMH) batteries are less susceptible to the 'memory effect' and can hold more charge for the same size. They are popular as rechargeable AA and AAA batteries. One down side of NiMH is they have a higher self-discharge rate than NiCd .

Lithium-ion (Li-ion) batteries are most preferred for devices like laptops and the newer cell phones and digital cameras. They have higher energy density figure, low self-discharge rate and quick recharge time compared to NiCd and NiMH. There are two types of negative electrodes used in Li-ion batteries: coke and graphite. Li-ion batteries clearly have many advantages but they also bring in certain complications in the device circuitry. They require protection circuits to ensure linear charge and discharge and to prevent overcharge and over discharged. If the user

overcharges the electrolyte solution is decomposed and gasses are released increasing battery pressure and lithium is precipitated causing a rise of fire and explosion. If over discharge occurs the electrolyte decomposes and the characteristics deteriorate.

In improving battery technology, there is greater focus on power management of these battery packs. Nowadays Li-ion batteries come equipped with a microchip that control charging, discharging and conditioning of the batteries. The power management system generally has circuitry for voltage detection, voltage regulation, switching regulation and battery protection (Shearer, 2007). For battery protection, it detects overcharge or overdischarge conditions and disconnects the battery. Overcharge is detected by over-voltage and overdischarge by under-voltage. Charging begins with checking for presence of good batteries. For good batteries if the voltage is low (below 3.2V) the charging is done with low current. Once the batteries reach a set value they are charged through maximum charging current. The discharge circuit generally has the cells in parallel with a voltage detection and voltage regulator to maintain constant output voltage.

2.4 Power Consumption Management

This section introduces the major concepts in power management. Three elements in power management are identified and discussed: mechanisms, policies, and architectures. All three elements are described in detail.

Power management is manipulating the components of a system to reduce power consumption. As already briefly stated in the introduction chapter, power management is important. For portable devices it will remain important for some time in the future because every improvement in power efficiency is countered by an increase in the number of features or performance, or in a reduction of the battery size and weight. We call this the power consumption balance principle. With a reformulation of the problem, the great importance of power management becomes clear. The improvements of features, performance, and reduction of the battery size and weight depend on power efficiency advances.

A prime example of the power consumption balance principle is the mass market of x86 laptops. Power efficiency improved with standards for power management such as Advanced Power Management, APMand Advanced Configuration and Power Interface, ACPI (Fisher et al, 2004) in combination with more efficient policies. However, the power consumption of laptops has not decreased over the years. When more efficient hard-disk policies are discovered, they are not used to reduce the average power consumption, but employed in the next hard-disk

model to increase storage capacity and to reduce access latency with compatible power usage. Every improvement in the power efficiency of hardware components themselves is also neutralized. In 1993, a typical display had no colour, a resolution of 640 by 480 pixels, and dissipated 2, but senior industry specialists predicted an aggressive improvement in battery lifetime. In the last two years, the battery lifetime of a typical laptop has increased from roughly 4 hours towards typically 8 hours due to technology advances in displays and other components. A decade later this prediction has proven to be false because consumers prefer a light laptop with a high-resolution colour screen instead of a doubled battery lifetime.

Power management solutions often divide the power management problem into more manageable pieces by treating each component in isolation. Furthermore, each component is managed by two entities: a power management mechanism and a power management policy. The policy determines the manipulation of a component and the mechanism contains the intelligence to do the actual manipulation.

Figure 4. The relation between a power management policy and mechanism (Pouwelse, 2003).

The relation between a mechanism and policy is illustrated in Figure 2. In the simplest case, a power management mechanism detects that a hardware component is idle and sends a sleep command to the power management mechanism of that component.

A more advanced approach is to be aware of trade-offs in the components and to exploit these to improve power efficiency. Such trade-offs can be made explicit through a utility curve. A utility curve of a component specifies the relation between performance or quality, power consumption, and other costs such as wireless spectrum usage. Each point on the curve represents an operational mode. The metric of performance or quality can be multi-dimensional. For example, the quality of Wireless communication is difficult to achieve compared to wired communication because of the time varying nature of the RF carrier. In general wireless communication is dominated by high error rates, varying signal to noise ratio, noise variations, limited bandwidth and multipath delays. These factors lead to retransmissions and effectively higher power consumptions.

Figure 5. The power usage with increasing performance or quality for three common utility curves, exponential (a), linear (b), and flat (c) (Pouwelse, 2003).

Figure 3 shows three common types of utility curves. Figure 3.a shows the exponential type of utility curve where low performance or quality means significant lower power consumption than an operational mode with high performance or quality. An example of such a utility curve is a processor that supports voltage scaling. The difference in power consumption per processor frequency can be very large. The linear type of utility curve is depicted in Figure 3.b; a performance increase results in a power consumption increase of equal proportion. A 802.11 wireless LAN card has such a utility curve; more bandwidth costs proportionally more power. Figure 3.c shows the utility curve of the flat type, where the activation of the hardware is costly and the subsequent usage cost is marginal. For example, rotating a hard-disk platter for one minute is relatively expensive in terms of power consumption; whether light or intensive data transfers take place during that time makes little difference.

In principle trade-offs such as processor performance versus power consumption are continuous, hence the name utility curve. However, a number of factors (e.g. product cost and design complexity) limit the number of operational modes in commercial hardware implementations. For example, Intel has limited the number of supported processor frequencies to just two with their SpeedStep technology (Intel, 2000). SpeedStep provides simply an economy mode and a performance mode for the processor, reducing the additional complexity and cost of the processor to a minimum.

There also exists an economic phenomenon that influences utility curves. This is called the mechanism/ policy interlock. In the highly segmented PC industry, hardware components are produced by a different company than software components. The hardware company is not strongly motivated to introduce new power-saving mechanisms when no policy exists to exploit them. On the other hand, a software company is only motivated to develop a policy for experimental mechanisms when a sufficient number of consumers have bought that hardware. This classical chicken-and-egg problem does not exist in single-purpose devices such as a cell

phone where a single vendor develops both mechanism and policy. Parallels exist with the problems of hardware / software co-design (Pouwelse, 2003) where the hardware (mechanism) cannot be independently developed from the software (policy).

The mechanism/policy interlock illustrates the close relation between mechanisms and policies. Policies and architectures are the main topic of this thesis, but without more mechanisms their efficiency gains will be limited. Future portable devices need both smarter policies and more mechanisms to increase power efficiency.

2.5 Summary

Batteries are the dominant technology for powering portable devices. Unfortunately, the possible influence of a run-time power management policy on battery performance is limited. Power management polices help in maximising performance and minimising power consumption. This implies that a policy must have knowledge about performance and power consumption, or resource usage in general.

Chapter 3

Information Security

3 Information Security

3.1 Introduction

The internet is not a single network, but a worldwide collection of loosely connected networks that are accessible by individual computer hosts, in a variety of ways, to anyone with a computer and a network connection. Thus, individuals and organizations can reach any point on the internet without regard to national or geographic boundaries or time of day.

However, along with the convenience and easy access to information come risks. Among them are the risks that valuable information will be lost, stolen, changed, or misused. If information is recorded electronically and is available on networked computers, it is more vulnerable than if the same information is printed on paper and locked in a file cabinet. Intruders do not need to enter an office or home; they may not even be in the same country. They can steal or tamper with information without touching a piece of paper or a photocopier. They can also create new electronic files, run their own programs, and hide evidence of their unauthorized activity. In This Chapter we discuss the Confidentiality, Integrity, Authentication, and Non-repudiation. Security protocols such as IPSec, WPA, and SSL are also discussed.

3.2 Network Security

Measures are taken in an organization to secure its data from attackers. The measures taken are generally not as simple as they appear to be. In developing the security of the system the designer has to look at the possible ways in which the systems security mechanisms would fail. The design of such systems also needs to consider where to place the mechanism both physically and logically. As explained in Chapter 1 the security services can be classified into the following: confidentiality, authentication, integrity, non-repudiation, access control, and availability.

Security services are designed to prevent attacks that compromise the security policy of the organization. The attacks may be passive attacks or active attacks. In the case of passive attacks the attacker monitors the network connections. By way of monitoring the connections the attacker can get the private information of the organization and can do traffic analysis in case

the content of the message cannot be decoded that easily. In active attacks the attacker modifies the communication in some ways to his advantage. Masquerade involves the attacker assuming the identity of someone else. Masquerade is thus an attack on the authentication service. Replay attacks involve the replay of information from previous valid connections. The replay attacks can be extended further by means of modification of the information. The information content is so modified that it appears to be from a legitimate source. Denial of service attack prevents or inhibits the normal use of the communications services. It involves disruption in the flow of information either by disabling the network resources of overloading them with meaningless data.

3.2.1 Confidentiality

Confidentiality is intended to prevent passive attacks. To make the information confidential, the data is modified in such a way that it would be infeasible for the attacker to guess the data. It is achieved by means of encryption algorithms. Encryption is done based on shared secret information between communicating parties. Only the receiver and in some cases the sender know how to decrypt the data after it has been encrypted. The data is generally encrypted with an encryption key and can be decrypted by using a decryption key. For a symmetric key scheme, the encryption and the decryption keys are the same. For public key schemes, they are different. The key used for encryption is called public key while the key for decryption is called the private key. Encryption is further explained in the next section. Confidentiality in some cases also requires hiding the process of communication itself to avoid traffic flow analysis. Raju Ramaswamy and Manual Mogollon, (Ramaswamy 1990, Mogollon 2007) explain different ways in which traffic flow analysis can be achieved at various levels of the OSI layer.

3.2.2 Authentication

In authentication services, it is required that a pair of communicating entities establishes its identity. Essentially, the authentication service tries to establish the identity by means of making sure that a secret is shared between the involved entities. Some protocols establish the authentication through the means of symmetric key schemes while others establish it through the means of public key schemes. For the users of a symmetric key authentication system the communication systems share a secret key between the two communicating parties. Authentication is generally achieved based on challenge and response procedure. A popularly used symmetric key authentication scheme is Kerberos. Kerberos was a part of the Athena Project at MIT (Stallings, 1999). Rather than building an elaborate authentication for each communicating entity, Kerberos makes use of a centralized authentication server.

Public Key schemes are slightly different from the symmetric key authentication systems. X.509 [Stalling 1999, Austin 2001) is one such public key authentication scheme. The use of public key schemes allows us to get rid of the dependency on the authentication server. However, the assurance of the public key is established by means of a certificate. In a certificate, trusted authorities endorse the validity of the user certificate by means of digital signatures.

3.2.3 Integrity

As for data integrity, assurance is needed that only legitimate entities can modify the message (Mont, 2004). Encrypting the message to some extent ensures that the attacker cannot modify the message. However there is a possibility of some malicious user sending random data to the receiver. The receiver would decrypt these messages to some incomprehensible data, which posses the possibility of some damage. One method of avoiding such situations is to add a checksum to the message before encrypting it. If the decrypted message and the checksum match then the received message can be assumed valid otherwise it is considered invalid. Such a scheme would provide authentication and confidentiality along with message integrity.

A variation to the use of checksums is the use of encrypted hash functions. A hash function takes a variable length of message, *M*, and produces a hash code, *H(M)*, of fixed size. The hash code closely depends on the message. Small changes in the message result in a completely different hash code. The hash codes are designed to have high collision resistance. This implies that given *H(M)* it is computationally infeasible to produce *M* or *H(M')* where *M'* represents some other message. The popular hash functions are MD5 and SHA (Hu et al, 2005).

3.2.4 Non-Repudiation

Non-repudiation involves the ability to prove to someone, the source of the document (Mont, 2004). The originator then cannot deny that he is the author of the document. In this sense non-repudiation involves both authentication and integrity. Symmetric keys are however inadequate in providing absolute non-repudiation even though they can provide authentication and integrity. The sender generates a hash code over the data and transfers it along with the data to the receiver. The receiver is able to check for the integrity of the document and it can authenticate the sender. However, the receiver cannot deductively assert that the data was sent by the se sender and that it was not modified by the receiver since the receiver also possesses

the same key and can generate the same hash and encrypt it. A reliable signature scheme with symmetric key scheme would require a trusted authority that can sign the document and check the document signatures.

With asymmetric key schemes non-repudiation can be achieved much more elegantly. Only the owner of the key knows the private key while any one can use the public key. Exploiting this fact has led to the evolution of the concept of digital signatures. Digital signatures allow the sender to generate a unique signature on the message that can only be generated by the owner of the private key. Everyone else including the receiver can verify the owner by using the public key but it is computationally infeasible for receiver to produce a similar signature for any other message.

The digital signatures make use of public key encryption and secure hash functions. A secure hash algorithm produces a hash value of the message. The hash value is then encrypted by the private key of the user using the public key algorithm. At the receiving end the receiver uses the public key to decrypt the hash value. It also generates a hash value from the message it has just received. If the hash value generated by the receiver and the received hash values match, the message is authenticated.

3.3 Wifi Protected Access (WPA)

The WPA and WPA2 standards were created by the Wi-Fi Alliance industry group that promotes interoperability and security for the wireless LAN industry. The Wi-Fi Alliance WPA and WPA2 standards closely mirror the official IEEE 802.11i wireless LAN security standards group but incorporate additional IETF EAP standards that the Wi-Fi Alliance considers secure. The WPA and WPA2 standards have two components (encryption and authentication) that are crucial to a secure wireless LAN. The encryption piece of WPA and WPA2 mandates the use of TKIP or, because it's considered to be more secure than TKIP, preferably AES encryption. From an encryption standpoint, WPA leaves AES optional while WPA2 mandates both TKIP and AES capability. The authentication piece of WPA and WPA2 before the Extended EAP update called for the use of a PSK (Pre-Shared Key) for personal mode and EAP-TLS for enterprise mode. After the Extended EAP update, there are now five EAP standards to choose from in WPA and WPA2 enterprise mode.

3.4 Internet Protocol Security (IPSec)

Virtual Private Network (VPN) provides the leverage to use the Internet as a private LAN. However, with the flexibility offered, they also expose the internal network to outside attacks. Hackers could access confidential data by exploiting accounts or by simply exploiting bugs in the access protocol. To prevent the problem of direct Internet access VPNs are often associated with mechanisms that provide authentication and encryption services. IP Security (IPSec) provides the means to implement security at IP level. It ensures security not just for applications that have security mechanisms but also to other ignorant applications that have not implemented any protection schemes.

IPSec provides security against open access, data manipulation, man in the middle attack and passive monitoring. IPSec has two functions by which it provides the security: Authentication Header (AH) to provide authenticity and Encapsulating Security payload (ESP) to encrypt the data portion of the IP packet (Stallings, 1999). IPSec allows systems to chose the algorithms and select the protocols used for services. The use of AH or ESP is defined by Security Association (SA). A SA is a one-way relationship between sender and receiver. Thus for two-way security two SAs are required.

The AH is based on the message authentication code (MAC). The MAC is generated by a shared secret between two terminal parties. The AH has a sequence number which is incremented every time the SA is used. This prevents replay attack. After all the sequence numbers for the SA are used a new SA needs to be established. The AH also has a value called the Integrity Check value (ICV). The ICV is a message MAC or a truncated version of a code produced by a MAC algorithm, which may be either SHA-1 or MD5. The MAC is calculated over the IP header fields that either do not change or have predictable change, AH fields other than the ICV field and the upper layer protocol data.

With Encapsulating Security Payload (ESP) the transmitter encrypts the payload of the IP packet. The data is padded to ensure the plaintext to be multiple numbers of bytes required by the encryption scheme. Number of algorithms can be used with ESP. These include the following: TripleDES, RC5, IDEA, CAST, and Blowfish. (Delfs, 2007)

Figure 6: IP Security Packet formats (Delfs, 2007)

IPSec can operate in two modes: Transport mode and the Tunnel mode. The Transport Mode provides protection primarily to the upper layer protocols. It is used for secure authenticated communication between two IP connected hosts. The ESP in Transport Mode encrypts the IP payload and optionally authenticates it while the AH in transport mode authenticates the IP Payload and selected portions of the IP Header.

In Tunnel Mode the complete original IP Packets are encapsulated inside the AH or ESP to provide a secure tunnel. Thus the original IP packet is treated as payload and is not examined in the intermediate stages. Tunnel mode is essentially established to create secure communication between two network endpoints, which could be routers or gateways. In tunnel mode the ESP encrypts and optionally authenticated the entire IP packet including the inner header. AH in tunnel mode authenticates the entire inner packet and selected portion of the outer IP header.

3.5 Secure Socket Layer (SSL)

SSL has been originally developed by Netscape and has been accepted widely by the World Wide Web for Internet secure communication between client and server (Stallings, 1999). SSL operates above some reliable transport protocol like the TCP layer that is the standard for connection oriented communications over the Internet. It comes below the application layer protocols like HTTP, IMAP. SSL facilitates the use of Server Authentication, Client Authentication and secure encrypted communication between the client and the server. SSL

fragments the message to be transmitted, optionally compresses it, applies message authentication code, encrypts it, and transmits the message using services provided by TCP. At the receiving end the messages are received, decrypted, verified, decompressed, reassembled and delivered to the upper layers (Brag et al, 2004).

A SSL session is an association between the client and the server. The SSL session protocol is layered and has two layers. One of the upper layers, the SSL Handshake protocol, allows the client and server to authenticate each other, negotiate the encrypting algorithm and the keys for encryption before the applications starts transmitting or receiving data. The upper layer is encapsulated by the SSL Record protocol. SSL session allows the use of multiple transient connections between the client and the server within on SSL session, which allows avoiding the renegotiations of the protocol parameters for each connection.

The SSL Record Protocol provides encapsulation to the upper layer protocols. It fragments the data into data blocks of 16384 or less. A lossless compression algorithm then compresses the fragmented blocks. The compression algorithm may be specified as null in case compression is not desired. The compression algorithm cannot expand the data above 1024 bytes in the worst case scenario. In the next step a message authentication code (MAC) is computed over the compressed data. The compressed message and the MAC are then encrypted using a symmetric key encryption scheme. The encryption may not increase the content length by more than 1024 bytes. The algorithm for MAC and symmetric key encryption can be selected during the handshake protocol. Initially the algorithm for MAC and symmetric encryption are null. (Brag et al, 2004)

Other than Handshake protocol there are two other upper layer protocols in SSL: Change Cipher Spec protocol and Alert protocol. The Change Cipher Spec protocol consists of a single message called the change cipher spec message that is used by the client and the server to notify the receiving party that the subsequent messages will be protected by the just-negotiated CipherSpec and keys. The Alert protocol is used convey alert messages to the peer entities. Alert messages convey level of severity and the description of the alert. The alert messages are encrypted and compressed like other messages. The following alert messages are included in the protocols: close notify, unexpected message, bad record MAC, decompression failure, handshake failure, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown, and illegal parameter.

3.6 Summary

The availability of lightweight, portable computers and wireless communications has made mobile computing applications practical. An ever more mobile workforce, home working, and the computerisation of inherently mobile activities are driving a need for powerful and complex mobile computer systems and applications integrated with fixed systems. Mobile cellular telephony is widely available and computers are being integrated with these telephones to form mobile computing devices.

Chapter 4

Encryption Algorithms

4 Encryption Algorithms

4.1 Introduction

Encryption forms the basic building block for various security services (Benatar, 2002). There are two types of cryptosystems: secret key and public key systems. In secret key schemes the same key is used for encryption as well as decryption. Most of the popular secret key algorithms are based on the Feistel Cipher Structure (Benatar, 2002). Encryption schemes like DES, IDEA, CAST, and AES use different kinds of transformation and rounds to achieve confusion and diffusion. In diffusion the statistical structure of the plaintext is dissipated into long-range statistics of ciphertext. Confusion on the other hand seeks to make the relationship between the ciphertext and the key as complex as possible. Although each one provides different mechanisms for encrypting data the basic security provided by the algorithm in today's context is decided by the brute force attack and is directly related to the key size.

Public key systems provide a radical departure from the secret key schemes. The public key scheme offered an elegant solution to the key distribution and authentication problems while using secret key mechanisms. Public key schemes are asymmetric involving the use of different keys for encryption and decryption process. They use mathematical functions known as the trapdoor functions to achieve encryption. The trap-door functions are based on some difficult mathematical problem. The IEEE 1363 (Goodman et al, 2001) document recognizes three distinct families of problems upon which the asymmetric key schemes can be based: integer factorization (IF), discrete logarithms (DL) and elliptic curves (EC). The popular schemes that use these methods are RSA, ElGamal and ECC respectively. Due to the computational cost of these public key schemes, they are generally used in conjunction with secret key schemes for secure data communications.

4.2 Symmetric Cryptosystems

The symmetric key algorithms also known as the conventional or one-key algorithms have the same key for encryption and decryption. For the communication between a sender and a receiver to remain secret the key should be kept secret. It can be denoted as:

$E_k(M) = C$

$D_k(C) = M$

where

E: Encryption function

D: Decryption function

M: Message

C: Cipher

k: Shared key

Symmetric Key algorithms can be classified into two categories: stream cipher that operates on a single bit at a time and block ciphers that operate on group of *n* bits at a time.

4.2.1 Stream Cipher

In stream cipher a bit stream, which is pseudo-random in behaviour, is generated from the key and is XORed with a stream of plaintext to produce the ciphertext stream (Li et al, 2007). The receiving end produces the same bit stream that is XORed with the ciphertext to recover the plaintext Stream Cipher encryption can be represented as:

$c_i = p_i \oplus k_i$

where

c_i: ith ciphertext bit

p_i: ith plaintext bit

k_i: ith key bit

Figure 7: Stream Cipher (Li et al, 2007)

Security of a stream cipher depends on the period of the bit stream produced by the keystream generator. Small periods lead to an insecure XOR operation. If the keystream generation

algorithm produces an endless bit stream we would have perfect security. In reality the period lies between the two extremes.

4.2.2 Block Cipher

A block cipher operates on plaintext block of n bits to produce ciphertext block of *n* bits. In substitution cipher there is an input to output mapping of plaintext and cipher text. The substitution cipher for small block size is however vulnerable to statistical analysis of the plaintext. A large block size would make the statistical characteristics of the result infeasible for cryptanalysis but is not practical. The key for such a scheme is the substitution itself hence for plaintext size of *n* bits the key size would be $n*2^n$ bits (Delfs et al, 2007)

Ciphertext

Figure 8: Feistel Cipher Scheme (Delfs at al, 2007)

Horst Feistel, a cryptographer who worked on the design of ciphers at IBM, proposed the concept of product cipher where two or more basic cipher functions are used sequentially such that the final product is cryptographically stronger than any of the basic ciphers. Feistel's proposal of a cipher that alternates substitution and permutations was actually an implementation of Claude Shannon's proposal to develop a product cipher that alternates confusion and diffusion functions; Confusion is basically defined as the concealment of the relation between the secret key and the cipher text, On the other hand, diffusion is regarded as the complexity of the relationship between the plain text and the cipher text. Shannon introduced the diffusion and confusion techniques to thwart statistical analysis on cipher text

(Delfs at al, 2007). The Feistel Ciphers achieve a reversible mapping between the plaintext and the cipher text based on a key by diffusion and confusion functions.

.

Confusion is basically defined as the concealment of the relation between the secret key and the cipher text, On the other hand, diffusion is regarded as the complexity of the relationship between the plain text and the cipher text.

In Feistel Cipher scheme the plaintext is split into two halves L_0 and R_0. The two halves pass through rounds of transformation and then combined to produce the ciphertext. Each round uses the output from the previous round and a sub key K_i derived from the main key K. Substitution is performed by round function F on the right half R_i of the data and then it is XORed with the left half L_i data. Interchanging the left and right halves of data performs permutation. The process of decryption is essentially the same as encryption. Here the cipher text is the input the keys are used in reverse order and the output is the plaintext.

4.2.3 CAST Encryption

Carlisle Adams and Stanford Tavares designed the CAST-128 encryption scheme (Stallings 1999, Shneier 1996). CAST- 128 has been published as RFC 2144 in May 1997. CAST has a classical Feistel network with 16 rounds and operates on 64 bits of plaintext to produce 64 bits of cipher text (Stallings, 1999). CAST makes use of keys that can vary from 40 bits to 128 bits. CAST Encryption scheme employs two subkeys 32 bit Km_i and 5 bit Kr_i in each round derived from the key. For key sizes less than 80 bits there are 12 rounds and for key size greater than 80 bits there are 16 rounds. Decryption is essentially the same with the key employed in reverse order. The structure of the F function used by CAST encryption scheme is given below.

Figure 9: CAST-128 Encryption Scheme (Delfs at al, 2007)

Round (i)	$f1_i$	$f2_i$	$f3_i$	$f4_i$
1,4,7,10,13,16	+	XOR	-	+
2,5,8,11,14	XOR	-	+	XOR
3,6,9,12,15	-	+	XOR	-

Table 2: Functions f1, f2, f3, and f4 in CAST based on rounds

Here <<< represents circular left shift operation based on the value of Kr_i. +,-, and ⊕ represent modulo 32 addition, subtraction and XOR. The 32-bit value obtained after the circular shift operation is then split into 4 8-bit inputs to *S*-boxes *S1, S2, S3,* and *S4*. The *S*-boxes take 8 bits input and produce 32 bit outputs. Functions *f1*, *f2*, *f3*, and *f4* play different roles in different rounds as listed by table 1.

There is no multiplication operation in CAST. The performance of CAST can be approximately summarized by the following equation.

$$\text{Processing time} = 21^* t_s + 22 * t_a + 21^* t_{xor} + 16 * t_{lshift}$$

Here t_s is time for subtraction, t_a time for addition, t_{xor} is time for XOR operation and t_{lshift} is the average time for circular left shift operation.

While implementing CAST the addition and subtraction operations can be easily achieved by using word size of 32 bits. Thus it can be considered equivalent to single operation. The shift operation may be available in processors like Pentium III and Celeron belonging to the IA-32 Intel® Architecture have these functions as single operations although the execution time may extend over several timing cycles. The *S*-boxes are implemented as two-dimensional arrays and the input data maps to columns and rows of the array.

4.2.4 IDEA Encryption

Xuejia and James Massey proposed International Data Encryption Algorithm (IDEA) encryption scheme in 1990 (Stallings, 1999). IDEA was designed as a stronger encryption scheme to replace the then existing DES encryption scheme. The IDEA encryption scheme consists of eight identical rounds of processing on the blocks and then a final transformation function.

The inputs to the scheme are 64-bit plaintext and 128-bit encryption key. The algorithm divides the input plaintext into four 16-bit blocks. Each round makes use of six 16-bit sub-keys to process the four 16-bit plaintext blocks and produces four 16-bit blocks. The final transformation round uses four sub-keys and has only the gray portion shown for a single round. In the *MA1* block, two of the four 16-bit blocks input to the rounds undergo modulo addition with two sub-keys and other two undergo modulo multiplication. The four intermediate 16-bits blocks are then combined to produce two 16 bit blocks which go as inputs to *MA2* structure. The *MA2* structure uses two keys to transform these two blocks using addition and multiplication operations and produces two 16-bit output blocks. These two blocks are combined with the intermediate four blocks using XOR function to produce the round output. In every round the second and the third 16-bit blocks are switched at the output to make the algorithm more resistant to cryptanalysis. The final transformation consists of only the *MA1* block of the rounds. This is done to make the algorithm symmetric so that the same code can be used for encryption and decryption with the keys applied in reverse order.

Figure 10: IDEA Encryption Scheme (Delfs at al, 2007)

Subkeys *k1-k8* are directly taken from the 128 bit key with *k1* being the first 16 bits to *k8* being the last 16 bits of the key. Then a 25 bit circular left shift is performed on the key and next 8 subkeys are generated. The procedure is repeated till 52 sub-keys are generated.

The process of decryption is essentially the same as encryption. The only difference is the input to decryption is the ciphertext and the sub-key generation is slightly changed. The first four keys are generated from the keys that were input to the transformation phase. The first and the fourth subkeys are multiplicative inverse of *k49* and *k52* while second and third subkeys are additive inverse of *k50* and *k51*. For every other round the subkeys derived are multiplicative and additive inverse of the corresponding 10-*i* round.

Computationally the most intensive operation in IDEA is the multiplication. Every round requires 4 16-bit multiplication operations in addition to 4 addition and 6 XOR operations.

$$\text{Processing time} = 34 * t_m + 34 * t_a + 48 * t_{xor}$$

Here t_m is time for single multiplication; t_a time for addition; and t_{xor} is time for XOR operation. For processors not capable of 32-bit multiplication the multiplication step can be improved by building a log table for the multiplication but it would take a lot of memory.

4.2.5 AES Encryption

The Rijndael algorithm designed by Joan Daemen and Vincent Rijmen was selected on October 2, 2000 by NIST as AES standard to replace the previous DES scheme for symmetric key encryption (Federal Information Processing Standards Publication, 2001). Although the original submission had provisions for variable length input blocks, AES takes 128-bits plaintext input and produces 128-bit cipher text output. The scheme can have three key sizes 128, 192 and 256. The key scheduling algorithm of AES takes the cipher key and produced *4 * (Nr + 1)* words, where *Nr* represents number of rounds. These words are called the key schedule words. The plaintext goes through 10 rounds for 128-bit key, 12 rounds for 192-bit key and 14 rounds for 256-bit key. All except for the last round are identical. Figure 8 below depicts the AES Encryption Scheme for 128 bits key.

In AES, the state word represents the two-dimensional array of bytes to be processed by the round. The bytes of the block are arranged in a two dimensional array of bytes. Since for AES the block size is fixed the bytes are arranged in two-dimensional array of four bytes and four columns. A 32-bit word maps to the columns of the state, where the most significant byte maps to the first row and the least significant to the fourth row.

Figure 11: AES Encryption Scheme with 128 bits key (Delfs et al, 2007)

The addition and multiplication operations in AES are slightly different from the conventional operations. The bytes are treated as polynomial where the bits represent the coefficients of the polynomial. Addition represents modulo 2 additions i.e., the XOR function. The multiplication is performed by polynomial multiplication modulo an irreducible polynomial of degree 8. For AES the irreducible polynomial is represented as:

$$m(x) = x^8 + x^4 + x^3 + x + 1$$

The modular reduction ensures that the result will be a binary polynomial of degree less than 8 and can be represented as a byte.

AES also makes use of four term polynomials of the form:

$$a(x) = a_3x^3 + a_2x^2 + a_1x^1 + a_0$$

Note that the polynomial here is different from the one before as the coefficients represent bytes of a word rather than bits of bytes. The multiplication of four-term polynomial is achieved but the addition and multiplication operations explained above and then reduce the result modulo a polynomial of degree 4. The polynomial for AES is $x_4 + 1$. The polynomial $x_4 + 1$ is not an irreducible polynomial in $GF(2^8)$; however, AES specifies fixed four term polynomial that do have and inverse.

The AES scheme employs 4 fundamental functions to achieve confusion and diffusion of data. The functions are SubBytes, ShiftRows, MixColumns, and AddRoundKey. SubBytes is a substitution table (*S*-box) transformation of the state. The *S*-box, which is invertible, operates independently on each byte of the state. In ShiftRows the rows of the state are cyclically shifted by *r-1* bytes, where *r* represents the row number. Thus the first row is not shifted at all; the second row is shifted by one byte; the third by two bytes and the fourth by 3 bytes.

In MixColumns the columns of the state are considered are four term polynomials and multiplied modulo $x_4 + 1$ with the fixed polynomial $a(x)$ given as:

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

In AddRoundKey the columns of the state are treated as words and are XORed with the key schedule words. Since it is an XOR operation AddRoundKey function is inverse of itself when applied again to the transformed word. For AES the function can be represented as:

$$[S'_{0,c}, s'_{1,c}, s'_{2,c}, s'_{3,c}] = [s_{0,c}, s_{1,c}, s_{2,c}, s_{3,c}] \oplus [w_{round*4+c}]$$

The decryption process is similar to encryption. Here inverse functions InvSubBytes, InvShiftRows, and InvMixColumns are used in place SubBytes, ShiftRows, MixColumns respectively. The InvShiftRows operated before InvSubBytes and AddRoundKey before InvMixColumns in the rounds. This assures that the order of operations applied during encryption process is applied in reverse.

The proposal of Rijndael, (Yan, 2002) provides details on the implementation of AES for a 32-bit processor. Accordingly different steps of the round can be combined in a single set of table lookups, allowing for very fast implementation. Each cell of the state can be separated and treated differently. Accordingly the transformations in the rounds can be expressed as

$$e_j = T_0[s_{0,,j}] \oplus T_1[s_{1,j-1}] \oplus T_2[s_{2,j-2}] \oplus T_3[s_{3,j-3}] \oplus w_{round*4+c}$$

Here T_0 to T_3 represent look up tables and e_j represents the j^{th} column of the output state. Other notations represent the same terms as explained above. Thus the AES implementation can be done by means of about 12 rotate byte operations and 16 XOR operations per round. Processing time can be summarized as follows:

Processing time for AES 128 bit key = 120 $t_{rotByte}$ + 164 t_{xor}

Processing time for AES 192 bit key = 144 $t_{rotByte}$ + 196 t_{xor}

Processing time for AES 256 bit key = 168 $t_{rotByte}$ + 226 t_{xor}

Here $t_{rotByte}$ is time for single rotate byte operation and t_{xor} is time for XOR operation.

4.3 Asymmetric Key Cryptosystems

Public key Schemes are called asymmetric key systems because they use two separate keys: one for encryption the data and other for decrypting it. The public schemes help a long way in

solving the key distribution problem of Conventional or Symmetric key systems. In symmetric key systems, the same key must be shared by the sender and the receiver and must be protected from others. Also, frequent changes in the key are advised to limit the amount of the data that would be compromised in case a key is revealed. Hence the security of the system is highly restricted by the security of the key distribution scheme. For Conventional scheme the possible way of key distribution is through physical transmission of the key or through the use of a key distribution center. The problem is further magnified by the fact that each connection should have a separate set of keys. This means that if there are *n* hosts communicating with each other than they would require *n(n-1)/2* keys for independent secure communication between hosts.

Public key schemes solve this problem. The structure of public key systems allows two different keys for encryption and decryption. Hence for a sender to communicate with a receiver, the sender uses the public key of the receiver to encrypt the data being sent. Only the receiver has the key to decrypt the data. Thus we no longer need *n(n-1)/2* key between n hosts and the number of keys required comes down to n. The receiver can publish his public key openly and any sender can use it to send secure data to the receiver. The receiver can at any time change its private key and publish the corresponding new public key in place of the old one to maintain the security of the system.

The question that comes to mind with this solution is why then should we use the conventional encryption systems. The reason why the conventional schemes are still very popular is because the public key cryptosystems due to their nature are very slow in encrypting and decrypting data. In fact, they are so slow that most of the modern day systems make use of a combinational scheme, which makes use of the public key schemes to exchange the secret key for the symmetric key scheme, which is actually used to transfer secure data.

A public key crypto system relies on non-deterministic polynomial-time algorithms. During encryption the message is converted from an easy instance to difficult instance through encryption key. Decryption converts the difficult instance to an easy instance by using the decryption key. The algorithms are so designed, given a difficult instance, the only way to find the easy instance without the decryption key can be best evaluated through non-deterministic methods. A non-deterministic algorithm involved guessing the solution and then verifying that solution to be correct. There is no known polynomial time solution for the problems and it is assumed that a polynomial time solution doesn’t exist (Kuo et al, 2002). In other words it is assumed that there is no deterministic solution to the difficult problem. Such non-deterministic problems can only be solved in exponential time. The RSA, ElGamal and elliptic curve cryptography systems’ way of using these mechanisms respectively have been explained below.

4.3.1 RSA Encryption

The Rivest-Shamir-Adleman (RSA) scheme is a block cipher scheme in which the plaintext and the ciphertext are integer between 0 and *n-1* for some n. The RSA scheme is developed using the exponentiation. In RSA, the keys are generated by selecting two large prime numbers *p* and *q* and their product is calculated as:

$$n = p * q$$

We calculate then Ø(n) as:

$$Ø(n) = (p-1) * (q-1)$$

Ø(n) is also called the Euler's totient function (Delfs et al, 2007), which is the number of integers less than *n* and relatively prime to *n*. Next a number, *e*, is selected is selected that is relatively prime to *Ø(n)* i.e., Greatest Common Divisor, GCD, of *e* and *Ø(n)* is 1. Then the multiplicative inverse of *e* is calculated using the Euclidean algorithm such that:

$$e*d = 1 \bmod Ø(n)$$

or

$$d = e^{-1} \bmod Ø(n)$$

The numbers *e, n* form the public key and *d, n* forms the private key. *p, q* and *Ø(n)* are discarded but never revealed. Size of *n* refers to the size of the key in RSA. *e* is generally chosen to be small, as the value of *e* is not known to affect the security of the scheme when proper encoding schemes are used (IEEE P1363, 2001). Typical values of *e* in use are *3, 17* and $2_{16}+1$. Although no particular attack with the context of RSA with Optimal Asymmetric Encryption Padding (OAEP) (RSA Laboratories, 2000) has been detected with the use of *e=3* the more conservative users prefer using public exponents larger than 3. The RSA problem can be solved fastest by the integer factorization method. Moreover systems can be designed to have a constant value for *e* so that *e* need not be transmitted. Size of *d* is approximately the same as the size of *n* (RSA, 2007)

To encrypt a message, the message is broken into small numbers, M_i, less than n. Let us call these message blocks. These numbers are raised to power *e* modulo *n* to obtain the cipher block, *Ci*:

$$C_i = M_i^{\,e} \bmod n$$

This is assuming the fact that it is mathematically difficult to determine M_i given C_i. To decrypt the cipher blocks are again raised to power *d* mod *n* to obtain the corresponding message blocks.

$$\begin{aligned} C_i^{\,d} \bmod n &= (M_i^{\,e})^d \bmod n \\ &= M_i^{ed} \bmod n \\ &= M_i^{1 \bmod \emptyset(n)} \bmod n \\ &= M_i \end{aligned}$$

Normally an encoding scheme is applied before encrypting the message for security purpose. The encoding scheme maps the message to and encoded message with some randomness that can be reversed for decryption of message based on the encoding parameters. The recommended encoding mechanism of RSA is Optimal Asymmetric Encryption Padding (OAEP) (RSA Laboratories, 2000).

Both encryption and decryption in RSA involve raising an integer to another integer mod n. Since the integers are large numbers, if the exponentiation is done before modulo operation the size of the intermediate result would be very large. To make it practical to implement the RSA algorithm the following property of modular arithmetic is exploited.

$$(a \bmod n) * (b \bmod n) = (a * b) \bmod n$$

Using this property along with successive multiplication scheme it is possible to compute x^e with less than *(e-1)* multiplications. For example x^{32} can be computed by computing the following intermediate results: x, x^2, x^4, x^8, x^{16}, x^{32}. Here the result could be obtained in 5 multiplications instead of 31.

Successive multiplication can be applied to any exponent and can be evaluated on an average in *1.5 * k* multiplications, where *k* is the size in bits of the exponent. For multiplication of to larger prime numbers with odd modulus the Montgomery Algorithm is used (Nedjah et al, 2003). The

runtime for the multiplication algorithm is proportional to $O(N^2)$ where N is the size in bits of modulus n.

Thus total run-time of encryption can be estimated as follows:

Processing time for encryption = 1.5 * k * $O(N_2)$

For decryption, the operation is made further efficient by using the Chinese Remainder Theorem (CRT) (Stallings, 1999). According to CRT, $M = Ci^d \ mod \ n$ can be calculated from the residues $Mp = Ci^d \ mod \ p$ and $Mq = Ci^d \ mod \ q$ since $n = p*q$. This allows reducing the computational time but reducing the modulus bit size to half.

4.3.2 ElGamal Encryption

The ElGamal Scheme (Schneier 1996, Stallings 1999, IEEE 2001, Cryptopp 2009,) is based on the discrete logarithm problem proposed by T. ElGamal in 1985. It is based on the Diffie-Hellman scheme. The ElGamal scheme uses randomization hence the same message block can produce different ciphertext block for a given public key.

For generating a key pair, first a large prime number p is chosen along with two other random numbers, g the generator and x, both less than p. Next we calculate y as:

$y = g^x \bmod p$

The discrete logarithm problem states that given y, and p it is computationally difficult to determine the value of x. The public key is y, g, and p. x is the private key. The key size refers to size of p in bits. For ElGamal scheme it is possible to use a generator g such that the group generated by g belongs to the subgroup of the order r. For details of groups, prime field and order of group refer (Stallings 1999, IEEE 2001). For the purpose of our discussion subgroup with order r represents the maximum unique modulo p numbers that can be generated by successive exponentiation of the generator g. r is a prime divisor of $p-1$ by definitions of subgroup. The private key x is chosen to be in the range 1 to $r-1$. The use of subgroup allows the use of short keys for exponentiation in ElGamal improving the overall performance of the scheme. The best known solution to the discrete logarithm problem is the Generalized Number Field Sieve (GNFS) (IEEE 2001) which has an asymptotic run time of $exp(((1.923 + o(1)) \ ln(p)^{1/3} \ (ln(ln(p))^{2/3})$ where $o(1)$ is a number that goes to zero as p increases. The exponent is thus

chosen such that the brute force on the exponent takes more time than GNFS algorithm. The sizes of exponent chosen for security purpose are as follows:

Modulus (in bits)	Exponent (in bits)
512	120
1024	164
2048	226

Table 3: Short Exponent size for ElGamal encryption and decryption (Crypto++, 2009)

To encrypt a message block, the message block is mapped to a number in the range 2 to *p-1*. Then a random number, *k*, in the range 2 to *r-1* is chosen (in case short exponents are not used *k* and *r* in the range 1 to *p-1*). We then compute the following two values

$$C1_i = g^k \bmod p$$

$$C2_i = (y^k * M_i) \bmod p$$

$C1_i$ and $C2_i$ form the ciphertext. To decrypt the message the following calculation is done:

$$\begin{aligned} C1_i^{-x} * C2_i &= g^{-kx} * (y^k * M_i) \bmod p \\ &= g^{-kx} * g^{kx} * M_i \bmod p \\ &= M_i \end{aligned}$$

The processing time for ElGamal for RSA is about the same equation as for RSA. However, in case of ElGamal it is possible to do pre-computation of values of the intermediate results: x^1 x^2, x^4, x^8, x^{16}, x^{32} for generator *g* and public key *y* for the successive multiplication operation based on *k* to speed up the process of encryption. This can be done because the value of the base, i.e, *g* and *y*, remain the same.

4.3.3 Elliptic Curve Cryptography (ECC) Encryption

The standardized Elliptic Curve Cryptography scheme, ECC, considered in this document is Elliptic Curve Integrated Encryption Scheme (ECIES). It is also Diffie-Hellman based scheme. The complete standard is specified in the IEEE P1363a draft (IEEE, 2001) . The underlying scheme is similar to DHAES scheme and makes use of basic cryptographic structures like

elliptic curve arithmetic along with symmetric key schemes, message authentication code, and cryptographic hash functions to achieve a hybrid asymmetric encryption scheme.

Elliptic curves are not ellipses but are called so because they are described by cubic equations similar to those used for calculating the circumference of an ellipse. Elliptic curves of use in cryptosystems are normally of the form:

$$y^2 = (x^3 + a*x + b) \bmod p \qquad ; \text{where } (4a^3 + 27b^2) \neq 0$$

Here, *a* and *b* are all real numbers and *p* is large prime number. For elliptic curves we are interested in points in the first quadrant from (0, 0) up to (p-1, p-1) that satisfy the mod *p* equation. Elliptic curves also include a point, *O*, called the identity point or zero point. The number of point on the curve is called the order of the curve and is represented as *#E*. A point, *G*, is selected on the curve which is the generator point. Let the order of this point be *r*. Order of a point is different from the order of the curve. The order is the minimum integer number with which the point has to be multiplied to obtain *O*, in other words, $rG = O$. The curve parameters are thus completely defined by *a*, *b*, *p*, *r*, and *G*.

4.4 Security levels of encryption algorithms

Determining the security of an encryption algorithm is a difficult task to do. In determining whether the algorithm is secure a cryptanalyst studies the mathematical characteristics of the encryption scheme to find shortcut methods of decrypting the ciphertext. Mathematical treatment to find flaws is generally hard and the shortcut solutions to break the schemes may not be discovered. In order to ensure that the algorithm is secure, the algorithms are shared in communities and forums for open public scrutiny.

The symmetric key algorithms described in section 2.1 are considered secure against differential cryptanalysis and linear cryptanalysis. The security of most symmetric key algorithms that have survived public scrutiny is evaluated in terms of size of the encryption key assuming that the best solution is only to do an exhaustive search on the possible keys. There are no known weaknesses of the algorithms. Daemen, (Hirani 2003) however discovered some 2^{51} weak keys in IDEA which if used for encryption could be easily detected and recovered. However the likelihood of these keys being selected out of the 2^{128} keys possible is very low and can be prevented in the implementation of the algorithm itself. IDEA is generally considered very secure and its theoretical basis has been widely and openly discussed since 1990. CAST has

been around since 1997 and the Rijndael algorithm used in AES was proposed in 1999. Table 4 below summarizes the characteristics of the symmetric key schemes discussed in this chapter.

Algorithm	Year Published	Key Size (bits)	Plaintext Size (bits)	Number of Rounds	Operations
IDEA	1990	128	64	8	34 mult + 34 add + 48 XOR
CAST	1997	88-128	64	16	21 sub + 22 add + 21 XOR + 16 circular shift
		40-80		12	16 sub + 16 add + 16 XOR + 12 circular shift
AES	1999	128, 192, 256	128	10, 12, 14	120 Rotate Byte + 164XOR, 144 Rotate Byte + 196 XOR, 168 Rotate Byte + 226 XOR

Table 4: Characteristics of Symmetric Key Encryption schemes

For RSA the best-known attack is based on integer factorization problem. If it is possible to factorize $p * q$, then the secret key d and be easily derived. The attack against ElGamal scheme is solving the discrete log problem. The best-known algorithm for factoring and discrete log problem is Generalized Number Field Sieve (GNFS) (IEEE 2001, Cryptopp 2009). The solution to the GNFS algorithm has an asymptotic run time of $exp(((1.923 + o(1))\ ln(n)^{1/3}\ (ln(ln(n))^{2/3})$ where $o(1)$ is a number that goes to zero as n increases and n is the modulus. This means the equation to obtain the same level of security as symmetric curve schemes can be written as follows:

$k = ((1.923 + o(1))\ ln(n)^{1/3}\ (ln(ln(n))^{2/3}$; k is equivalent size of symmetric key

For elliptic curves the best attack known is the Pollard-ρ algorithm. The asymptotic runtime of the algorithm is $O(\ q^{1/2}\ )$ where q is the order of the curve. IEEE 1363 (IEEE, 2001) states that the order of an elliptic curve is approximately equal to the prime p used for modulus. This implies size of be p should be at least $2 * k$ for security equivalent to k bits of symmetric key scheme. Table 5 below summarizes the relative strength of keys required for different types of algorithms. These are rough estimates taken from (Ferguson, 2003) based on the best-known solution to the problems on which the encryption is based.

Table 5: Key Sizes recommended for Security (Ferguson, 2003)

It can be seen that the size in bits of RSA/DL schemes increases at a much rapid pace compared to elliptic curve schemes. The increasing key size leads to an increase in the size of the numbers being used in the system and hence the computational cost of operations on them. For the same reason is anticipated that elliptic curves without performing RSA and ElGamal in the near further. Hence a lot of attention is being paid towards elliptic curve as a viable alternative for public key cryptography.

4.5 Summary

Chapter 4 discusses encryption schemes used in this research and draws comparisons between encryptions in symmetric, such as stream cipher, block cipher, DES, IDEA, CAST and AES. It also discusses and asymmetric cryptosystems such as RSA, ElGamal, and ECIES, and explains the differences between these schemes with reference to time and power.

Chapter 5

System State Security Management (SSSM) Framework

5 System State Security Management Framework

5.1 Introduction

When designing secure, extensible applications for mobile devices two crucial questions must be answered:

1. What types of security policies can be expected to be enforced by the system, i.e. the device operating system?
2. What class of mechanisms are needed to enforce these policies?

Neither of these questions can be answered successfully without deep understanding of the area of enforceable security policies and the power of various enforcement mechanisms.

The first major effort to define the category of enforceable security policies is due to Schneider, (Schneider, 2001). Schneider examined the security traits that can be enforced by a specific type of program monitor. These monitors can introduce itself between an un-trusted program and the device on which the program runs. Schneider monitor can observe the series of security-relevant program actions one at a time, and if it identifies an action that will breach its policy, the monitor terminates the program. This method is very common since decisions about whether or not to terminate the program can depend upon the entire history of the program's execution. But, since these monitors can only terminate programs according, it is possible to define more powerful application aware security system.

In this chapter, we examine the question of which security policies can be enforced at run time by monitoring the system state. Our overall approach differs from Schneider's, who used automata to model program monitors, in that we view program monitors as transformers that edit the stream of security policies to respond to various system conditions.

5.2 A System State Security Management Framework

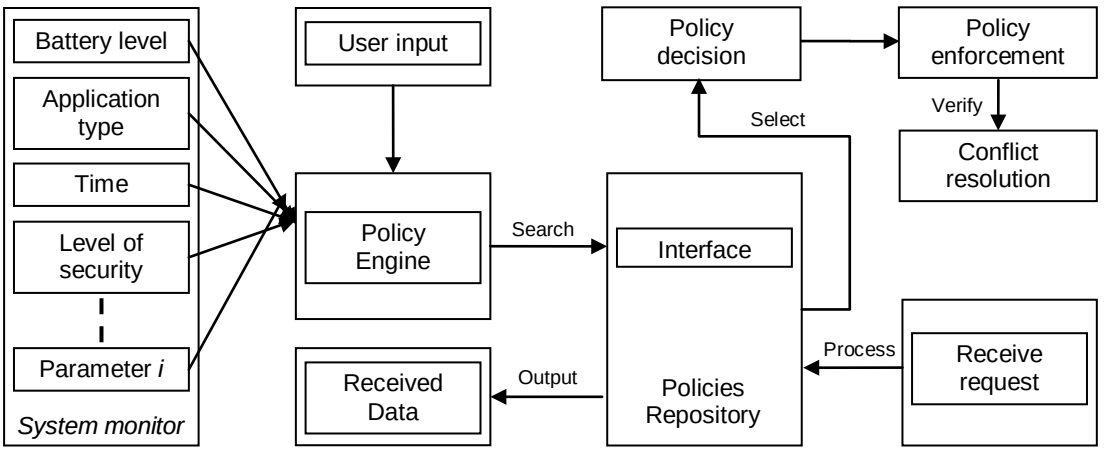

Figure 12: A component diagram of the SSSM framework.

This section researches dynamic reconfiguration of security policies in the context of dynamic adaptation. Various mobile device system parameters are classified to have effect on dynamic adaptation of security policies that differ in power and complexity. Therefore, we argue for an integrated framework that implements several system condition parameters and leverages Policies Repository compatible benefits by applying the appropriate one in a given situation.

The percentage of mobile devices offering open platform functionality is likely to continue to rise as time moves on. Open platforms puts forward great advantages all elements of the mobile system by bringing the ability for flexible program and service delivery choices that may be installed, removed or restored many times according to the user's needs and requirements. Nevertheless, open platforms allows unrestricted access to mobile APIs and resources by applications of un-trusted or unknown origin that could result in damage to the user application, the device, the network or all of these, if not managed by suitable security architectures and network security measures. Currently, mobile devices security is provided by most open OS mobile devices (Symbian OS, Microsoft, BREW, etc.). Recently, few industrial groups have also proposed recommendations including the GSM Association and Open Mobile Terminal Platform (OMTP).

The challenging security and privacy demands of new portable devices are forcing fundamental approaches to solve complex security management problems. This section presents a framework called System Monitor Security Management (for short, SSSM) which collects information about various system entities and dynamically optimise the security policies accordingly. When a user starts an application on a mobile device, SSSM collects system information such as the available battery level, the application type, the required level of security, amongst others. The collected information is fed into the Policy Engine module that uses complex algorithms to select a suitable policy or set of policies, e.g. encryption algorithm, to provide differentiated treatment to programs according to the system state. The Policies

Repository module stores the system generalised security policies that represents a continuum of security-management policies and allows to dynamically adapt the policy in run-time manner, e.g. gradually shifting from highest security performance to minimum as the battery depletes. The chosen security policy/policies are applied on all outgoing data streams to insure the desired level of protection. During an application session, the same security policy applied on outgoing data streams is applied on received data to produce information in a particular format that can be utilised by the application or interpreted by the user.

SSSM has the capability to adapt running security policies on run-time when a system state parameter changes. During SSSM execution, system state parameters can be changed continuously during the system-run. SSSM receives dynamic system state updates and respond to changes that has a significant effect on the chosen security polices performance.

In SSSM, one application session can have one or more security policies that mange the application events. For example, when a user uses his mobile device to login into his network operator to buy calling credit security policy #1 can be used to manage the customer account login information exchange and security policy #2 can be used when the customer proceeds with the payment. In this example security policy #2 provides higher level of protection as sensitive information, i.e. credit card information, is being transmitted.

The System Monitor module can contain any number of parameters that describe the system conditions. In this thesis we restrict our study to few important system conditions such as the energy and required application security levels. The chosen parameters are the most significant parameters in a mobile device and application running on it. These parameters will be used to evaluate the performance of the SSSM framework.

In the following sections we discuss the components of the SSSM framework in greater detail.

5.2.1 System Monitor

This is an essential component of the SSSM framework. It contains a set of parameters that precisely describe the current state of the system. These are parameters of the hardware platform, e.g. battery level and available memory, of the mobile device as well as the software components of the system, e.g. application type. System state is monitored at all times during the application run and various parameters are kept up-to-date. This component can contain any number of parameters that all/subset of them can be fed into the Policy Engine module to chose a suitable security policy. As the number of parameters utilised by the Policy Engine module

increases, the complexity of the selecting security policies increases. Therefore, in order to reduce the calculation and computation complexity, only those parameters that has effect on the choosing a suitable security policy are considered by the Policy Engine module. Information collected in this phase will determine the type of attack and the required security policies to prevent such attacks.

Some of the important system state parameters are:

1. Battery level: energy is a scarce and valuable resource for a mobile device, and managing it more effectively results in increased battery life, and, in the end, a competitive advantage. Computation comes second after communication in energy consumption so it is always desired to reduces the computation complexity e.g. the number of encryption rounds.

2. Application type: The classic challenge: security must be balanced against flexibility. Too much security and the system will be too inflexible for the type of rapid reaction required by today's business environment. Depending on the specific requirements and the type of application, an application security assessment should typically consist of the discovery of information on the type of environment that exists in the mobile device and that at the server side as well (e.g. embedded SQL queries specific to a single database version).

3. Time: time is an important factor that must be dealt with carefully. Some applications require rapid response time and long delays are not acceptable. Time is a quality of service metric that includes the processing time at the Policy Engine module and the execution time of the chosen security policies. The time complexity of a problem is the number of steps that it takes to solve an instance of the problem as a function of the size of the input, using the most efficient known algorithm.

4. Level of security: different applications have different levels of security requirements. Level of security refers to processing data with different security clearances in the same system or network.

We stress that the system can take any set of parameters but we restrict our framework evaluation to the above listed parameters.

5.2.2 Policy Engine

A Policy Engine is a software component that executes one or more security policies in a runtime production environment. A runtime environment allows applications to invoke security policies stored in the Policies Repository and execute them using the Policy Engine. The main function of the Policy Engine is to interpret stored policies, to construct the filters that are required, to implement the policies that map to the current system state and satisfy the application requirements. The improved efficiency of application security through increased policy selection automation results in increased control over implemented security policies for compliance and better security management.

The Policy Engine module detects and monitors the current conditions of the system software components as well as that of a host device hardware platform. Based on the detected and monitored conditions, the Policy Engine module coordinates security policies on the host device by configuring on-the-fly a plurality of functional polices supported by the host machine for performing various security related tasks. When monitored conditions permit a different configuration of security policies, the Policy Engine module on-the-fly reconfigures the deployed security policies.

This component has access to the Policies Repository component to select the policy/set of policies that need to be applied for a specific action. Security polices selected by the Policy Engine can be based on user input besides the information provided by the System Monitor module. This allows the user input his preferences and makes a selection between two equal options. User intervention is always required by the some device and/or applications to avoid situations such as a malicious application sends messages to premium services and run up a huge phone bill without the user detecting it. For example, when an internet connection is required most GSM mobile phones provide three options for the user to choose from: (1) Not allowed; (2) Ask every time; (3) Always allowed.

5.2.3 Policies Repository

Security policies are stored in the security provider database called “Policies Repository”. The security policy might come from legal regulation, company policy, or other sources. A security policy is composed of three components: Policy Conditions, Expressions, and Policy Statements. A policy condition is a condition under which a security policy will be executed.

These policy conditions, along with the specific information the user supply for the condition (such as user name, or security role, or start/stop times), are called expressions. A policy statement is the collection of expressions that define who is granted access to a resource, and is therefore the main part of any security policy. The ability to use multiple expressions means that the user can create complex security policies that meet his application security requirements. The use of `and` and `or` between these expressions, as well as the ordering of the expressions, is also an important feature to define compound policies.

This module contains a number of encryption algorithms that are suitable for various application types and provide different security levels. Different security algorithms can be used to encrypt different application data depending on the data sensitivity. Chapter # provides a detailed study of the encryption algorithms studied in this thesis. Other security polices such as scan file for virus before download or forbid connection to websites that present a security threat can also be included in this component.

The Interface is provided as a component of the Policies Repository which provides the ability to: register, define, classify, and manage all the security polices, verify consistency of policy definitions, define the relationships between different policies, and relate some of these polices to applications that are affected or need to enforce one or more of the policies.

5.2.4 Policy decision and enforcement

The policy decision module is separated from the policy enforcement module. Since the decision module should provide a conflict resolution plan used by the policy enforcement module. During the system lifecycle, initially consistent global policy may lead to unenforceable or inconsistent policies. Therefore, the policy enforcement needs to verify the selected policy rules against the security capabilities of the policy enforcement module. There are several relevant papers devoted to different techniques of conflict detection and resolution such as deontic logic (L.Cholvy, et. al.), dynamic conflict detection and resolution (N.Dunlop, et al.), policy conflicts specification and resolution (Morris Sloman, et. al.), detecting conflict of duty (D.Ferraiolo, et. al.), credential-based approach to specification of access control policies, conflict resolution in event-based policy management (Jan Chomicki), etc.

Applications security covers action taken throughout the application's life-cycle to stop exceptions in the security policy of an application or the underlying mobile system (vulnerabilities). Notice that applications only manage the use of resources assigned to them,

and not which resources are assigned to them. They, in turn, resolve the use of these resources by users of the application through application security policies.

In SSSM, all applications transitions must comply with the dynamically defined security policies.

5.2.5 Receive Request

Many applications that run on mobile devices interact with external software components or communicate information across the network. For example, an email application can access the dictionary software that is located on the mobile device; after that the message is sent over the internet. All communications sessions between the local application and the server for example are performed under an integrated security policy in both directions. Since, it is possible to have different policies to control different application transitions, all outgoing and ingoing requests/messages are analysed under the security policy created for that session, e.g. use the same encryption algorithm to decrypt a message at the other end of a communication session.

5.2.6 Received Data

This is the output of processing all received requests/messages after processing under the security policies that corresponds to that interaction session. An example is a decrypted message that an application received that needs decryption before further processing is possible.

5.2.7 Policy enforcement

In asynchronous distributed environments, enforcing appropriate security policies is more challenging task than defining and selecting of these policies. This is because security policies often depend on the actual information exchange among the distributed entities (the mobile device and the server), the order of events is not always defined, amongst other factors. In this section we study the difficulty in translating local policies to the entire distributed environment. We propose a mechanism for the enforcement of global security policies. The focus is on local events on specific node that has influence on the global state of the whole system.

Different security events occurring on the mobile device or the server may move the whole system from a legal to an illegal global state. Both the server and the mobile device run a local instance of the monitor. Each monitor performs local enforcement based on the local security

state and policies defined by the system owner. The local system state is passed to the server that incorporates the state of the remote device to create a new global security state which combines the states of both nodes. For simplicity, we assume that a simple communication model in which the security mechanism is capable of detecting all the communication channels. We also assume that a mobile device communicates with the server only. If the device is to communicate with any other device, this happens through the server.

Let P be an enforceable security policy that can be enforced by the local security system called Slocal. Slocal is used as input for the construction of the global security environment Sglobal. If a system has n parameters, then the system can have up to 2^n states (powerset construction). Thus, the Aglobal is based on the local state of the mobile device and the server. Let st be the state of the system at certain time. The global state at time t is denoted by Stglobal. Vector clocks (Mattern, 1989) are used to define the casual ordering of the security states. Each node maintains and updates its vector clock as described in (Mattern, 1989). Here, only security related events triggers vector clock update operations.

When a local event happens on a device, the Policy Engine performs several operations. First it updates the system vector clock, and then it updates its state. If an event is undefined, then the latest action violates the system policy and the transaction (e.g. send/receive) is aborted. Otherwise, system generates a message that contains its state and vector clocks. The receiver of the message extracts the received information and updates its vector clock by selecting the maximal value for every entrance from both vector clocks. Therefore, the local vector clock associated with that particular channel represents the combined knowledge of both nodes and both nodes update their local states.

However, sometimes, the nodes disagree on the negotiated security policy or the negotiated policy may contain inconsistencies. In this case, the nodes enter a second round of negotiation to resolve any conflicts.

5.3 Experimental evaluation

The experiment was conducted by studying the most common encryption algorithms used, symmetric algorithms such as IDEA, AES, and CAST, and asymmetric algorithms such as RSA, AlGamal, and Adobe leverages (RSA 512-1024, 2048). Crypto++, free C++ class library, was used as it has implementation for most of the popular cryptographic algorithms. It has implementation for AES, IDEA, DES, Triple-DESRC2, RC5, Blowfish, Diamond2, TEA,

SAFER, 3-WAY, GOST, SHARK, CAST-128, Square, Skipjack, Panama, ARC4, SEAL, WAKE, WAKE-OFB, BlumBlumShub, RSA, ElGamal, Nyberg-Rueppel (NR), Rabin, Rabin-Williams (RW), LUC, LUCELG, DLIES (variants of DHAES), ESIGN, ECIES. The dedicated author Wei Dai actively maintains the library.

In our research we used Cryptoo++ library and concentrated on evaluating IDEA, AES, CAST, AlGamal, RSA, and ECIES. Each algorithm was simulated on Windows XP based Toshiba SA30-203 P4 laptop, this enabled us to check all processes running for each algorithm, observing memory statistics and diagnosing the system logs in order to achieve a reflective indication of algorithms' behaviours under dynamic environment. These results were analysed and interpreted to give us an adequate indication of battery power usage to compare it with the security level provided by each algorithm. The encryption libraries used in the experiments were treated as separate classes and not part of the library application to reduce the running time. However, the libraries can be optimised by assigning a fixed number of iterations as well.

It is expected that the computation time would be closely related to the battery requirements; however, since the CPU utilization of power depends on parameters like voltage supply and capacitive load. The capacitive load on the CPU depends on the switching demand, which again depends on the instructions being executed. Hence, measurements for both the parameters are considered.

The battery capacity keeps reducing with the number of times it is charged and discharged. Hence, care is taken during the experiments that the battery is not unnecessarily discharged too many times. Also to ensure that this effect does not affect the results, the measurements for a group of readings for one section are taken together. In all about 100 discharge cycles have been completed on the laptop would lead to reduction in capacity of 5% for Li-ion batteries that the laptop uses from the characteristics published by Toshiba.

The experiments note the number of iteration or runs over the file and the battery life. So to restrict the number of measurements statistical properties were gathered by taking the difference in battery life left. Change in battery life divided by the number of runs gives the battery life consumed in percentage for one run. It was observed that since the charge and discharge cycles of the laptop were more or less linear and the memory effect that leads to non-linear discharge characteristics is negligible for Li-ion batteries.

To demonstrate how the calculations were done consider Table 6 as an example of the observations for a particular scheme.

Observation Number	Number of Iteration	Percentage Battery Left
1	10	98
2	20	97
3	30	95
4	40	94
5	50	92

Table 6: Sample table of observations

Next we calculate the difference in iterations between different observations as follows.

Observations	Difference of Iteration (a)	Change in % battery left (b)	Battery Consumed Per Iteration (b/a)
1,2	10	1	0.1
2,3	10	2	0.2
3,4	10	1	0.1
4,5	10	2	0.2

Table 7: Sample table for calculations

The average battery consumed per iteration can thus be calculated as follows

$$\text{Average battery Consumed per iteration} = (0.1 + 0.2 + 0.1 + 0.2)/4 = 0.15$$

Thus the statistics for reliability was gathered by taking every battery life change divided by runs to get battery life per run over the period of the experiment.

5.4 Comparison of algorithms

Purpose of this section of experiments is to see the trade-off in choosing secure algorithms and their computational and battery requirement costs.

5.4.1 Symmetric Key Schemes

We have reviewed of symmetric key schemes: AES128, CAST128, IDEA in chapter 2. All of these schemes provide 128 bits key encryption schemes and provide the same level of security as discussed earlier in section 2.2.3. However they have different performance in terms of time and battery requirements. Crypto++ library provides implementation of AES, CAST and IDEA.

Programs are written with each scheme that record battery drain and time after encrypting a 5MB file 10 times. Also a case where just the file is accessed and written to another file is considered to differentiate between the file access load and encryption load. In case of No Encryption and AES128 the minimum data block size accessed is 128 bits (16 bytes) while for CAST128 and IDEA the minimum data block size accessed is 64 bits (8 bytes).
In secure communications architecture it should be useful to change the key size based on the battery power remaining in the system to reduce the load. Hence effect of varying key size to 128, 192, and 256 bit keys is studies for AES algorithm.

AES has 10 rounds for 128-bit key. It should be possible to change the number of rounds of encryption to make a quality to battery utilization compromise. Reducing the number of rounds would make the algorithm less secure but would have the advantage of being able to use the same key and adaptively switching the number of rounds based on the channel and the device battery status.

5.4.2 Asymmetric Key Schemes

We have reviewed asymmetric key schemes: RSA, ElGamal, ECIES in chapter 3. Programs are written with each scheme that record battery drain and time after encrypting a 5MB file 10 times. Also a case where just the file is accessed and written to another file is considered to differentiate between the file access load and encryption load. In case of RSA the test is further extended. The effect of varying key sizes from 512 bits to 2048 bits is studied.

5.5 Summary

This chapter introduced a novel system state-aware framework that aims to automatically adapt security mechanisms according to the current state and conditions of various software and hardware components of the mobile, wireless system. SSSM is made up of many components; some components collect information about the current status of various hardware components and application requirements that is then used by another component to adjust the system behaviour to achieve optimality. The performance of the SSSM framework is experimentally studied in the following chapter.

Chapter 6

Laptop Results

6 Results

6.1 Introduction

We consider various symmetric and asymmetric key schemes with different key sizes. The graphs for the results obtained are plotted in this section. The performance of the schemes and their implications has been analyzed in this chapter and suggestions have been made for designing energy efficient secure communication systems. The appendix section can be checked for actual numerical values that were obtained during the experiments.

6.2 Symmetric Key Schemes

The figure 13 below shows the battery consumed and the time consumed for iteration over a 5MB file. The comparison is done between AES encryption, CAST encryption and IDEA all with 128 bits key size using Crypto++ library. The 'No Encryption' scheme represents the case where the file was just accessed and the data from this file was put into another file. This was included to see actually how many extra resources are put into encrypting over accessing the data.

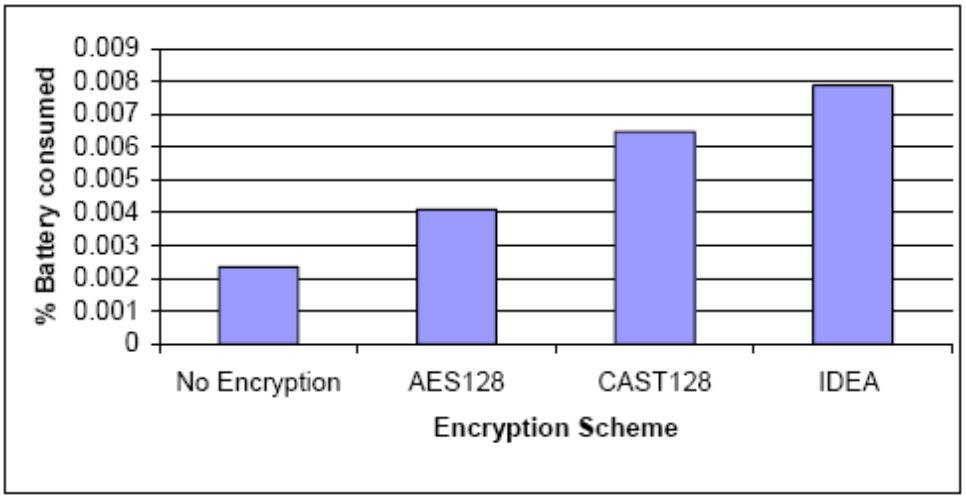

Figure 13: Percentage Battery Consumed by symmetric key schemes.

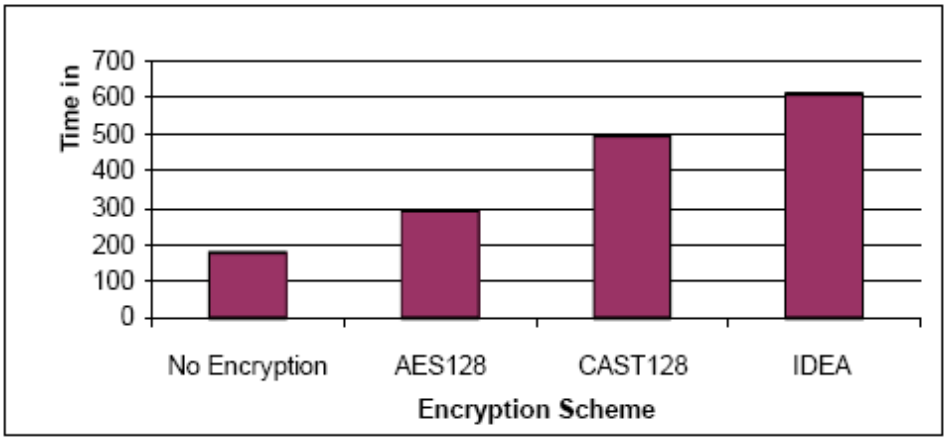

Figure 14: Time Consumed per iteration by symmetric key schemes

From the figure 14 we see that CAST128 and IDEA are more power hungry than AES128. The reason for this is that the algorithms employ different diffusion and confusion techniques. If we recall the implementation details of the AES, CAST and IDEA algorithms it can be seen that the major computationally intensive operations in them were 120 rotate bytes, 21 circular left shift and 34 multiplication operations per encryption assuming that addition, subtraction and XOR operations are relatively less intensive. CAST128 and IDEA process 64 bits at a time while AES128 processes 128 bits at a time. The time and the battery consumed are proportional to the complexity of operations per round and the number of data moves required making it difficult to state how much effect each type of operation has in the performance of the schemes. This is because data moves are difficult to analyze with different register allocations that can be done by the compiler and state of the operation system.

Encryption of the plaintext data with AES128 causes increase in battery consumption by 75% and time consumption by 65%, when compared to the No Encryption scenario. CAST128 consumes 58% and IDEA consumes 92% more battery than AES128. CAST128 takes 70% and IDEA takes 110% more time than AES scheme. The total energy of the Toshiba battery is 58Wh.

An immature approach to calculate the energy consumed per iteration over the file would be to multiply the percentage of battery consumed to 58Wh. For example, battery consumed by AES128 would be *0.00499 * 58 = 0.2377 Wh* for processing 5 MB of data. In terms of security, there are no known attacks on the unmodified version of CAST and AES. IDEA however has a set of 2^{51} weak keys (Delfs at al, 2007) hence AES and CAST can be considered more secure than IDEA.

6.2.1 Key Size variation

We compare the change in performance by using different key sizes for AES algorithm. For AES we consider the three different key sizes possible i.e., 128 bit, 192 bits and 256 bit keys.

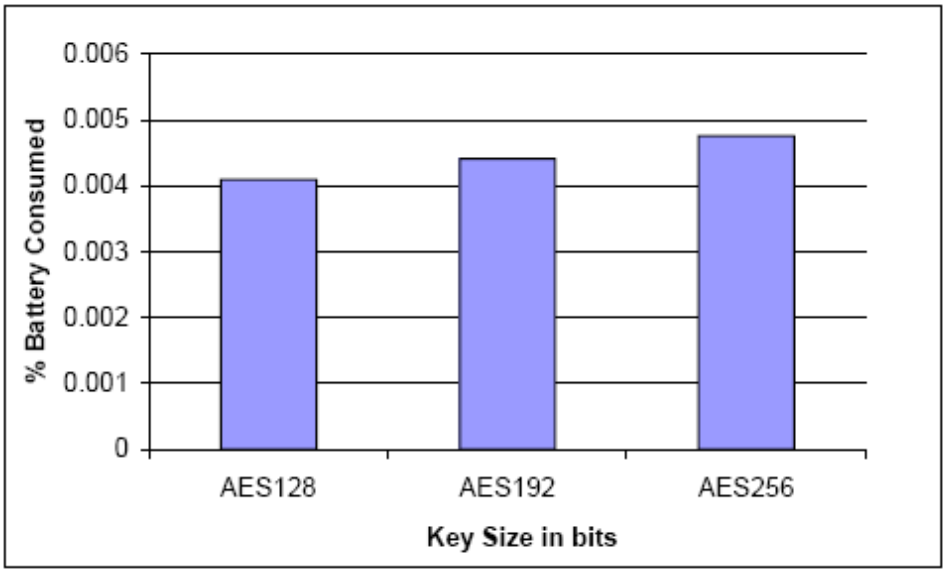

Figure 15: Percentage Battery Consumed with different Key Sizes for AES

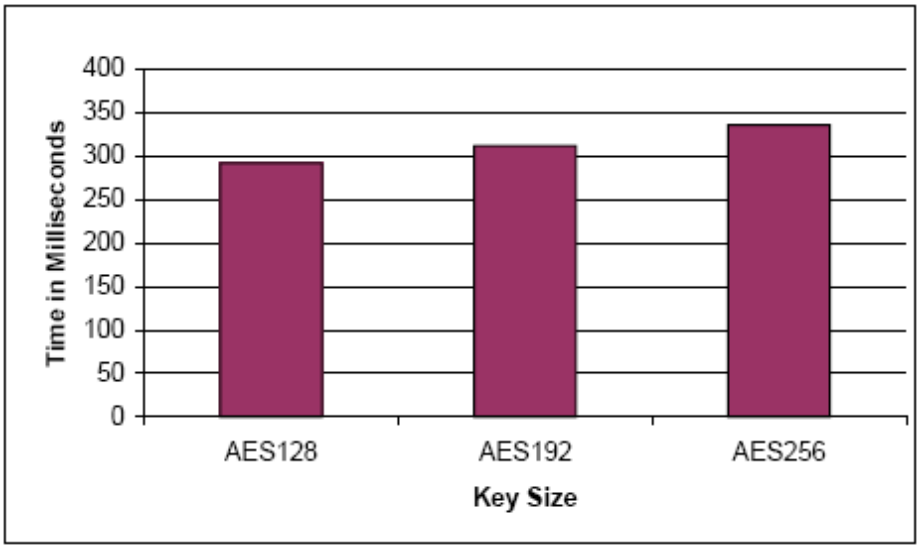

Figure 16: Time Consumption with Different Key Sizes for AES

Increased key size leads to increased security as shown in table 4. In case of AES it can be seen that higher key size leads to noticeable change in the battery and time consumption. It can be seen that going from 128 bits key to 192 bits causes increase in power and time consumption by about 8% and to 256 bit key causes an increase of 16%. AES128 has 120, AES 192 has 144, and AES256 has 168 rotate byte operations, which implies 20% and 40% more operations for AES 192 and AES256 when compared to AES128. Although there seems an increase in power consumption that is directly proportional to the increased operations, the increase is less amplified. This can be attributed to the fact that the data access from the file over which the

operations are performed has already been carried out. However, the increased power consumption of higher key size posses a compromise that should be considered before choosing the size of the key. For normal application 128 bits key is considered very secure hence going for higher key sizes would mean unnecessary wastage of resources for the added security that is actually not required.

6.2.2 The effect of Changing the Number of Rounds

The AES encryption scheme has 10 rounds for 128 bits key. It should be possible to reduce the number of rounds so that the amount of battery and time consumed while encrypting the data could be reduced. Figures 16 and 17 below show the comparison of energy and time consumed by the reduced round version of AES 128 bits key encryption.

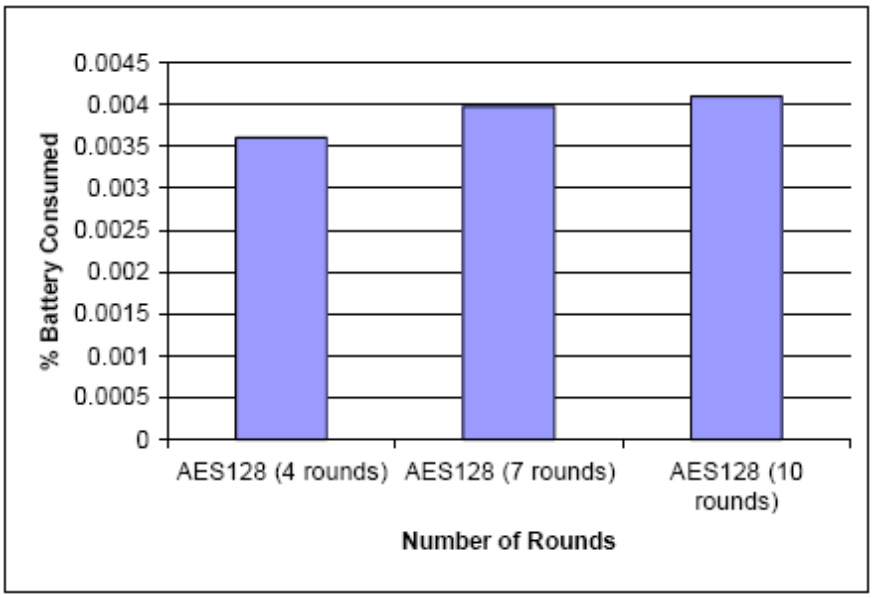

Figure 17: Percentage battery consumed by different number of rounds for AES 128 bit-key

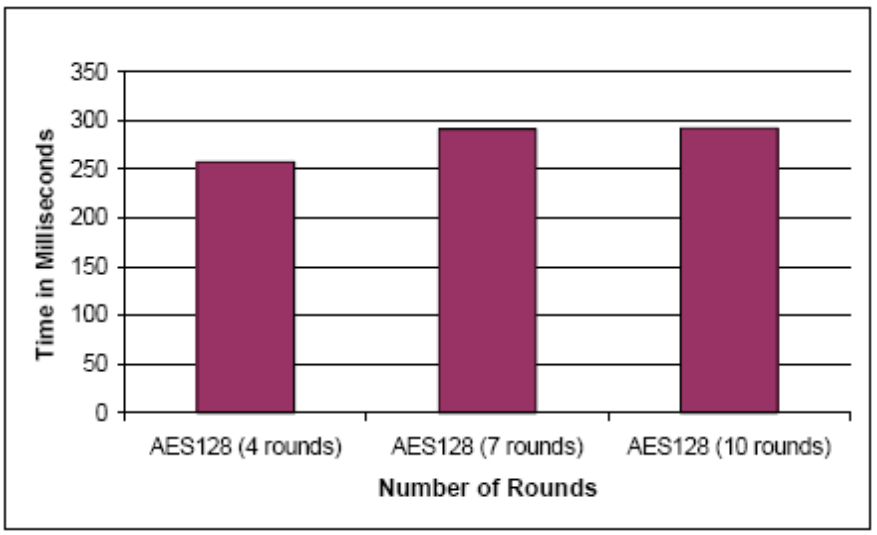

Figure 18: Time Consumed by different number of rounds for AES 128 bit-key encryption

As can be seen from the graph by reducing the number of rounds from 10 to 4 rounds it is possible to save 13% of battery and time consumption and from 10 to 7 rounds is 3%. Reducing the number of rounds would decrease the security of the encryption. The NIST report on AES provides the work factor or operations and the memory required to attack the scheme under reduced number of rounds for 128 bits key size. No known attacks have been reported against the actual AES scheme with 10 rounds and the only way is through brute force which would take 2^{127} encryptions on an average.

Rounds	Operations	Memory
4	2^{9}	Small
5	2^{40}	Small
6	2^{44}	$7*2^{32}$
7	2^{120}	2^{61}
10	2^{127}	Not Known

Table 8: Attacks reported on reduced round variants of AES with 128 bits key

It can be seen that 4 round encryption with AES is not very secure and under most cases would not be preferred. Compromising the number of rounds leads to highly reduced security for 4 rounds. As the number of rounds increase the security of the scheme improves, as can be seen from the table 8 above. There is likelihood of more attacks on the reduced round version of AES being discovered which increases the risk in using these schemes. The designer should consider the level of security required and the battery saving resulting from the performance improvements by reducing the number of rounds with extreme caution.

6.3 Asymmetric Key Schemes

In asymmetric key scheme RSA, ElGamal and ECIES are considered. The ECIES and ElGamal are hybrid schemes used in combination with symmetric key scheme in this case AES. The implementation of RSA used has OAEP standard for encoding then message with SHA for hashing. The RSA standard specifies the asymmetric key encryption but doesn't specify how it can be used in conjunction with symmetric key schemes. The implementation is open to user. ElGamal scheme like RSA is also just encryption. It does not use any encoding or hash functions. ECC used here has XOR for symmetric key scheme and SHA (Stallings 1999, Delfs 2007) as hash function. The secp160k1 curve provided by Certicom is considered for ECIES. For the purpose of the performance measurement the encryption is considered in units of 16 bytes of data. The ElGamal and ECIES encryption schemes employ pre-computation with

storage space of 16 locations. In the case of Asymmetric key schemes the file size considered is 1MB instead of 5MB as the processing with asymmetric keys is much more time consuming.

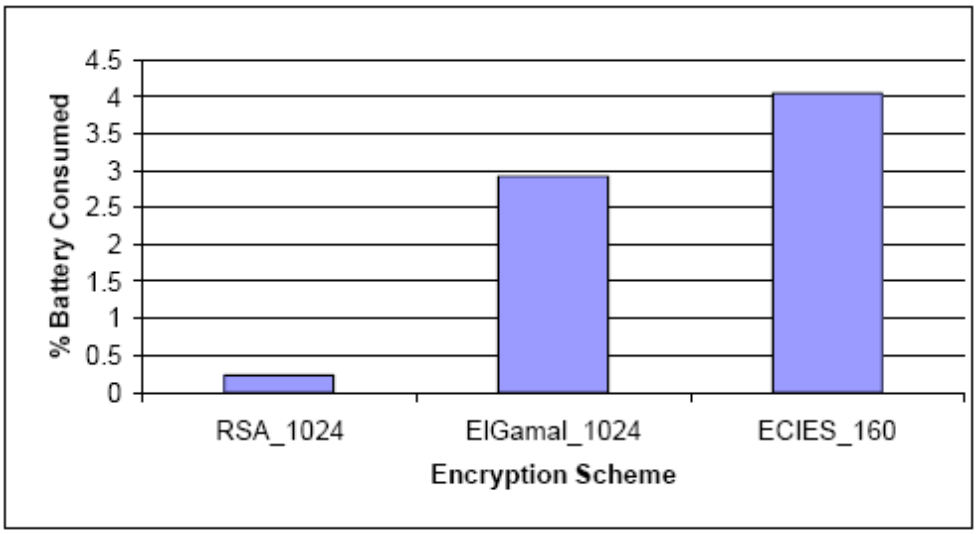

Figure 19: Percentage Battery Consumption of Asymmetric Key Schemes

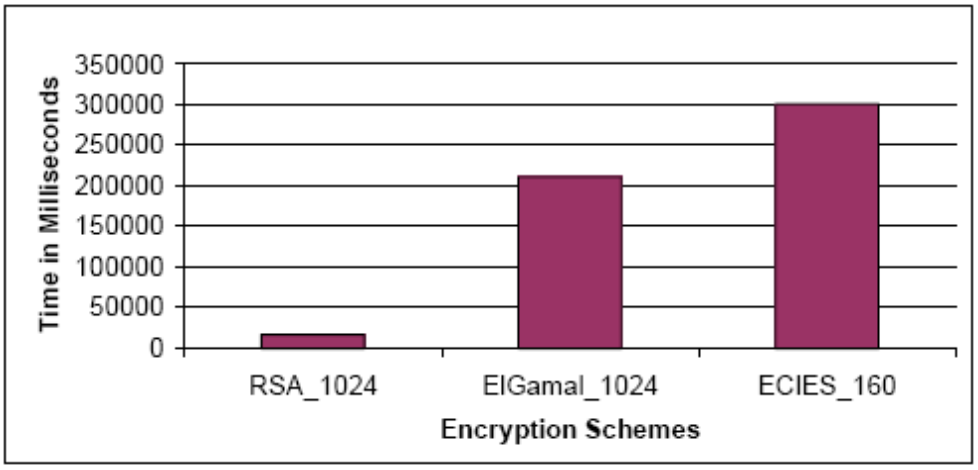

Figure 20: Time Consumption of Asymmetric Key Schemes

The figures 19 and 20 above show the performance of the schemes in terms of the time and the battery consumed without data transmission. We can see that for encryption, RSA is much more efficient than the ElGamal and ECIES schemes. In fact, RSA 1024 bits encryption is as much as 17 times better than the ECIES 160 bits curve and 12 times better than the 1024 bit ElGamal scheme. RSA encryption is much faster than the other two schemes because it uses a short key for encryption, which is much smaller than the short exponent used in ElGamal. The exponent for RSA used in these experiments is 17 and can be represented in 5 bits while the short exponent in ElGamal is 164 bits. ElGamal encryption requires two exponentiation operations while RSA requires one. Also, the 160 bits elliptic curve multiplication in ECIES is costlier than the exponentiation of the short exponent in ElGamal. The assumption while making the statement is that the encoding schemes have a relatively low impact computational requirement as the RSA and ECIES schemes employ SHA functions before encrypting the message.

In our experiments it is also essential that we consider decryption performance of the algorithms. Figure 21 below gives the graphs for decryption process of these algorithms.

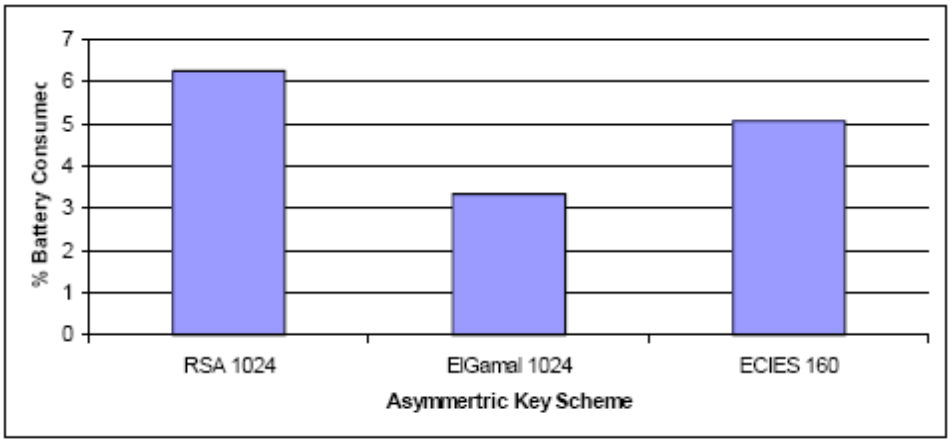

Figure 21: Percentage Battery Consumed by Asymmetric key decryption

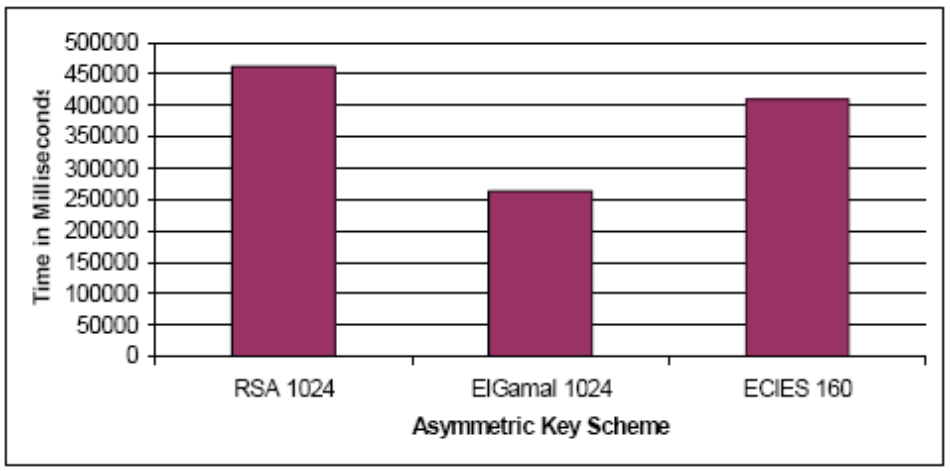

Figure 22: Time Consumption of Asymmetric Key Decryption

It can be seen that in the decryption process ElGamal 1024 is the fastest scheme. ElGamal decryption requires only one exponentiation operation with a 164 bit short exponent. RSA also employs one exponentiation for decryption but it does not employ short keys, as short decryption key would make RSA insecure. The exponent in RSA is 1024 bits and in ElGamal is 164 bits. Even after application of the CRT for RSA, which reduces the modulus size to 512 bits, ElGamal is still more efficient. ECIES decryption requires two elliptic curve multiplication operations. Also ECIES 160 bits decryption is about 20% more efficient than RSA 1024 decryption.

Looking at the performance of the Asymmetric Key Schemes RSA is very efficient for encryption while ElGamal is the most efficient for the decryption process and should be chosen to save the energy consumed. ECIES is the most energy hungry of all the schemes when energy

consumption with both encryption and decryption is considered. However, RSA and ECIES used here are more secure than ElGamal because they employ message authentication functions to ensure that messages decrypted are not invalid.

6.3.1 Key Size variation

We compare the change in performance by using different key sizes for RSA algorithm. The next figures show the performance of different key sizes of RSA. 512, 1024 and 2048 key sizes are considered. For encryption the file being encrypted was 1MB and for decryption the decrypted file was of 1MB.

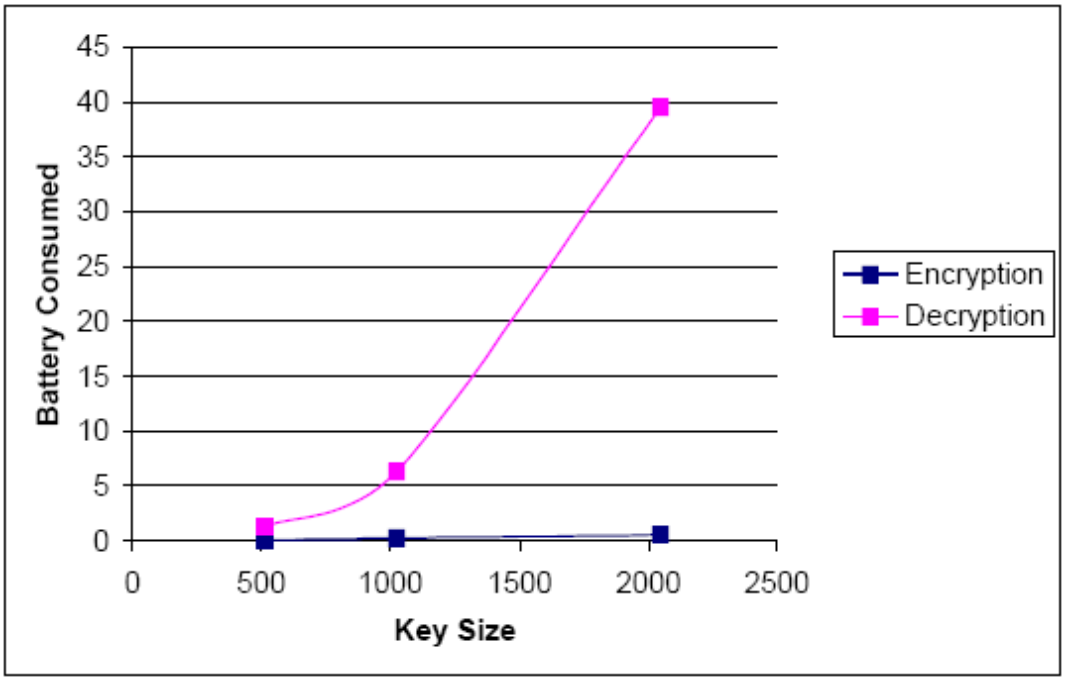

Figure 23: Percentage Battery Consumed with different Key Sizes for RSA without data transmission

Increasing the security by increasing the key size in case of RSA has a higher cost. The curve shows that as the key size doubles the battery and time consumption requirements are more than double indicating a nonlinear growth in consumption with increase in key size. In Chapter 2 we have seen that the Processing time for RSA is proportional to $1.5 * k * O(N^2)$, where k is the exponent and N is the size of the modulus. The exponent remains constant for encryption and the processing time is proportional to $O(N^2)$. However, for decryption the exponent depends on the key size and follows a third order growth proportional to $O(N^3)$ as $k \alpha N$. Choosing 2048 bits security over 1024 bits security results in three times more energy consumption for encryption and 6.6 times more energy consumption for decryption. By current standards 1024 bit key is

considered secure for normal applications until 2013. 2048 need only be chosen for highly secure applications.

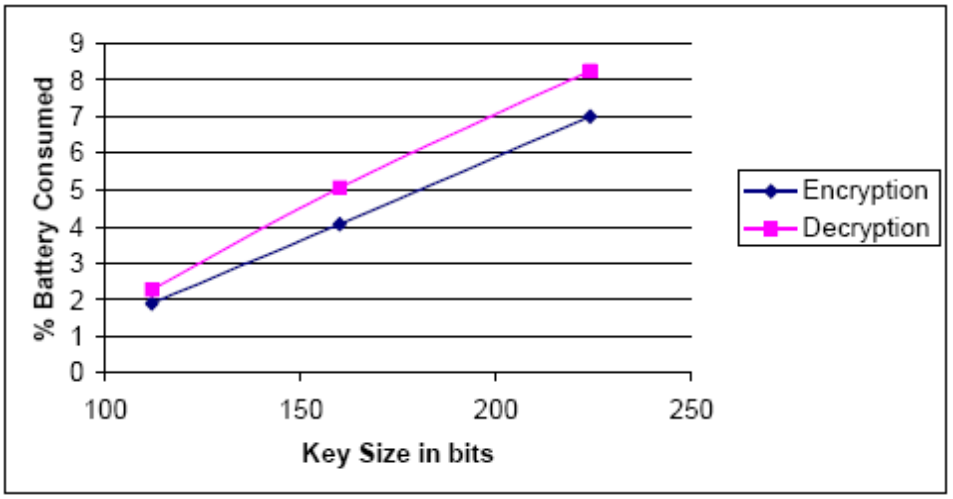

Figure 24: Percentage Battery Consumed with different Key Sizes for ECIES without data transmission

The figure above shows the increase in battery consumption with increasing key size. For encryption the file being encrypted was 1MB and for decryption the decrypted file was of 1MB. The curves used for ECIES are the ones defined by Certicom. For 112 bits secp112r1 is used, for 160 bits secp160k1 is used and for 224 bits secp224k1 is used. There is increase in battery consumption by 114% with 160 bits key and 270% with 224 bits key.

Security Year	RSA			ECIES		
	Key (bits)	% Battery (Encrypt)	% Battery (Decrypt)	Key (bits)	% Battery (Encrypt)	% Battery (Decrypt)
1982	512	0.1047	1.2727	112	1.8918	2.258
2013	1024	0.2372	6.2727	160	4.05	5.0769
2055	2048	0.5833	39.5	224	7.0000	8.25

Table 9: Comparison of percentage battery and time consumed by RSA and ECIES for different key sizes

112 bits key of ECIES is equivalent to 512 bits key of RSA and 224 bits key is equivalent to 2048 bits key of RSA. We have seen that ECIES decryption with 160 bits key is more efficient than RSA with 1024 bits key. For higher security requirements ECIES proves to be much more efficient for combined encryption and decryption performance. This is so because the key size grows much gradually for ECIES the key size of RSA for the same increase in the level of security.

6.4 Summary

This chapter shows the results obtained by testing AES, IDEA, CAST, RSA, ElGamal, and ECIES on Laptops. The differences between the mentioned encryptions are evaluated with reference to time and power consumed in both decryption and encryption stages.

Chapter 7

Handheld Devices

7 Handheld Devices

7.1 Introduction

The Pocket PC and Handheld PC are considered in this experiment. The size of the file processed is 1 MB since these devices have much lower resources than the laptop.

Figure 25: Percentage Battery Consumed by symmetric key schemes without transmission on Pocket PC

On the Pocket PC, CAST128 consumes 22.14 % more power than AES128. Also encrypting data with AES128 is 50% more costly then just processing plaintext data. AES is still more efficient than CAST. AES256 consumes 9% more energy than AES128 and AES192 consumes 5% more energy. RSA 1024 encryption is 29 times more costly than AES 128 encryption the Pocket PC. The results for RSA were shown separately in a table for better comparison of the symmetric key schemes in the graph.

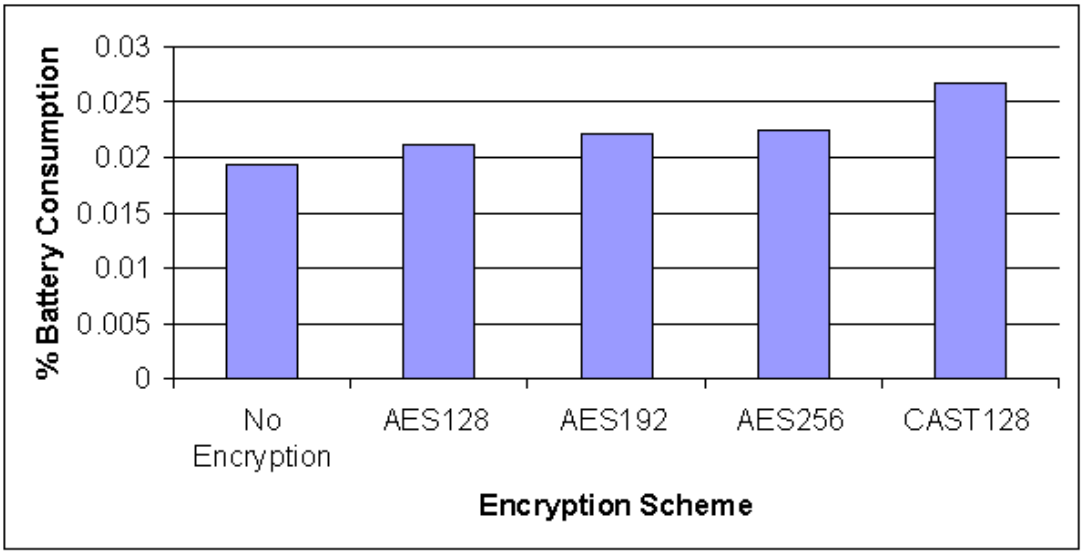

Figure 26: Percentage Battery Consumed by symmetric key schemes without transmission on Handheld device

On the Nokia 9300, CAST128 consumes 26% more battery than AES128. Encryption is 10% more costly than access just accessing data. AES192 consumes 4% and AES256 about 6% more battery than AES128.

Encryption	Satellite1200 (laptop)		IPAQ HP3870 (Pocket PC)		Nokia 9300	
	% Battery	Time (ms)	% Battery	Time (ms)	% Battery	Time (ms)
No Encryption	0.0004678	35.3016	0.008502	1632.53	0.019363	2253.16
AES128	0.00082	58.3904	0.012755	2391.46	0.021205	2266.93
AES192	0.0008826	62.4436	0.013367	2348.32	0.022182	2313.89
AES256	0.0009526	67.2373	0.013860	2391.46	0.022429	2358.4
CAST128	0.001293	99.4639	0.015578	2748.45	0.026782	2729.58
RSA 1024 (encrypt)	0.2372	17039.9	1.944444	361459.46	2.273268	357280.0
RSA 1024 (decrypt)	6.2727	462858.09	27.5	5114000	33.5	4453000

Table 10: Performance of Encryption Schemes on Laptop, Pocket PC and Handheld

When compared to a laptop encryption 1MB on the laptop with AES128 would consume 0.00082% battery, which implies that on the Pocket PC AES128 is 16 times costlier and 26 times costlier on handheld Nokia 9300. The comparison made here is just in terms of the percentage battery and not the amount of energy consumed. The energy consumed on each device may give very different results but for practical applications the percentage battery consumed figure should be a much more relevant figure in making a decision about system design. In terms of time consumed Pocket PC takes 41 times and handheld takes 38.82 times

more time to encrypt with AES128 encryption. The time consumed by handheld and Pocket PC is nearly the same. This is because the IPAQ was used with the expansion pack, which effectively provides the IPAQ with more battery power while that is not the case with the Handheld computer.

Without the expansion pack the performance is expected to be similar for Pocket PC and Handheld device. The Nokia 9300 has a 206 MHz 32-bit StrongARM SA1110 processor with 32 MB SDRAM that operates at 51 MHz with display area of 82.5 square centimeters. IPAQ HP 3870 also has a 206 MHz 32-bit StrongARM SA1110 processor but with 64 MB SDRAM at 100 MHZ and a display area of 48 square centimetres. It can be seen that handheld has about twice the display area as Pocket PC and half the memory with the same processor. The access to data in the Handheld computer in much slower than that of the Pocket PC because of the slower RAM and hence there is a difference in the performance of the data access with no encryption scenario.

7.2 Wireless Environment

We consider the effect of changes in the communication environment by changing packet sizes, signal to noise ratio, and layer where encryption occurs.

7.2.1 Data Transmission

Figures below show comparison graphs for the same schemes with transmission of the data over the network. It has been noticed that the size of the data unit transmitted per unit time affected the results hence to remove that factor of variability from consideration two units of data were combined in the case of CAST and IDEA so that same amount of data is transmitted for all the scheme i.e., a data unit of 16 bytes was maintained for all schemes.

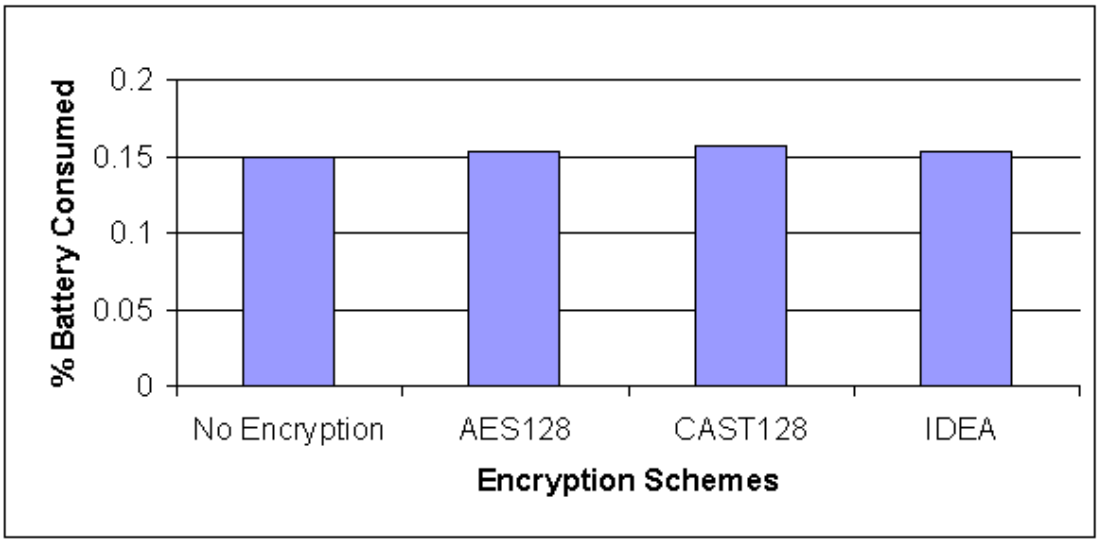

Figure 27: Percentage Battery Consumed by symmetric schemes with data transmission

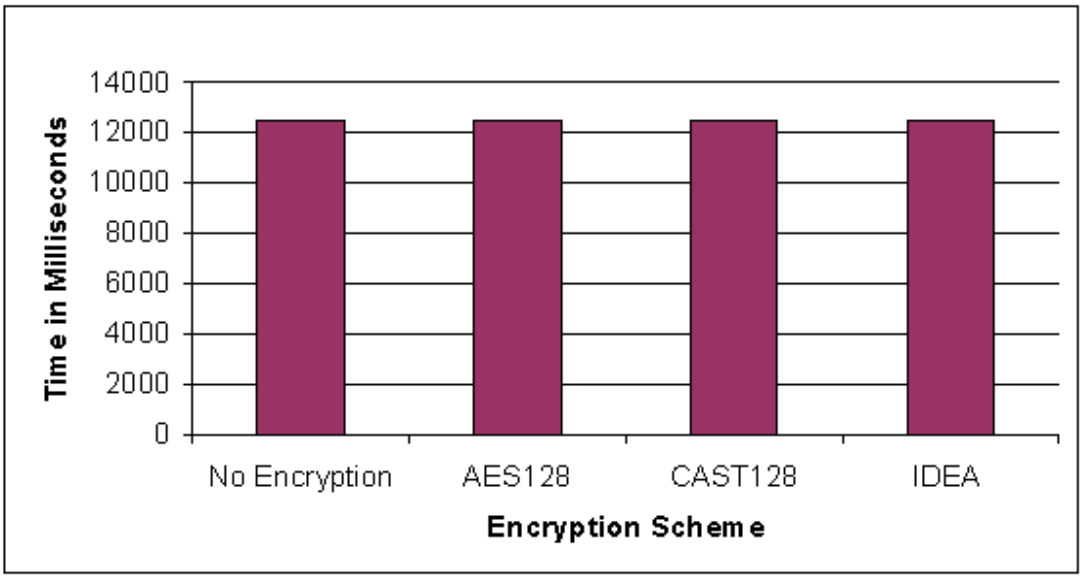

Figure 28: Time Consumed by symmetric key schemes with data transmission

The above graphs show the battery and time consumed per iteration over a 5MB file. From the chart it may be concluded that the encryption scheme used makes almost no significant difference in the energy and time consumption of encryption schemes. It can be seen that if we consider the 90% confidence interval for each of the case above we cannot say that either of the case is better than other. Most of the energy and time goes into transmission of data rather than encryption and since encryption and transmission are dependent on different resources the task that takes max time and resources dominates the results i.e., in this case transmission.

The significance of the particular algorithm used would only come into picture when there are other computation intensive processes running parallel with the application using secure communication at application level, which by the data should be negligible. It is estimated that in a multi-threaded environment when the contention for computation resources reaches a stage

where data transmission can become faster than data access and encryption the importance of the encryption scheme used will be realized. In any case AES128 appears to be a better choice.

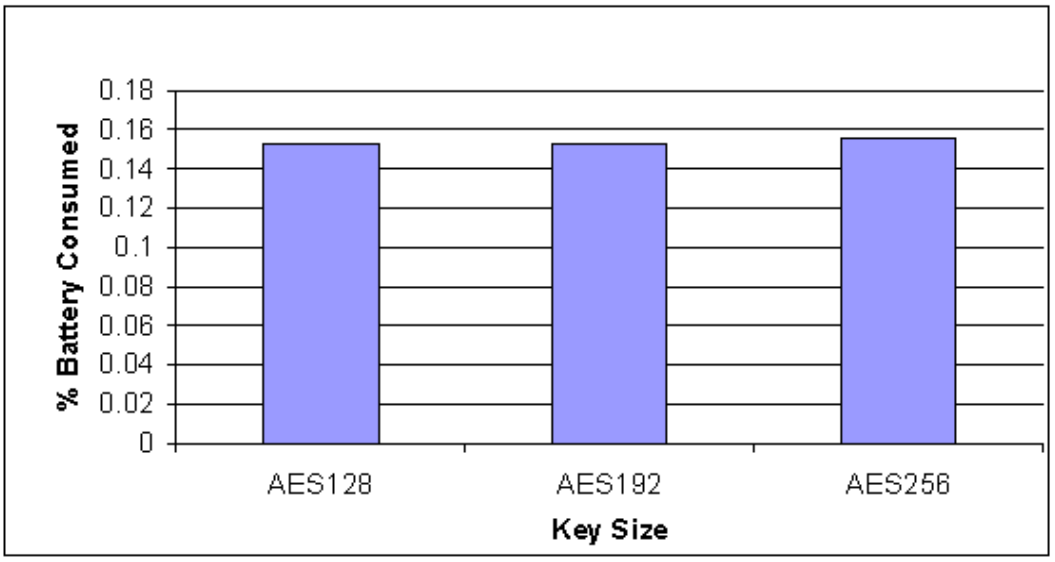

Figure 29: Percentage battery consumed by different AES Key Sizes with data transmission

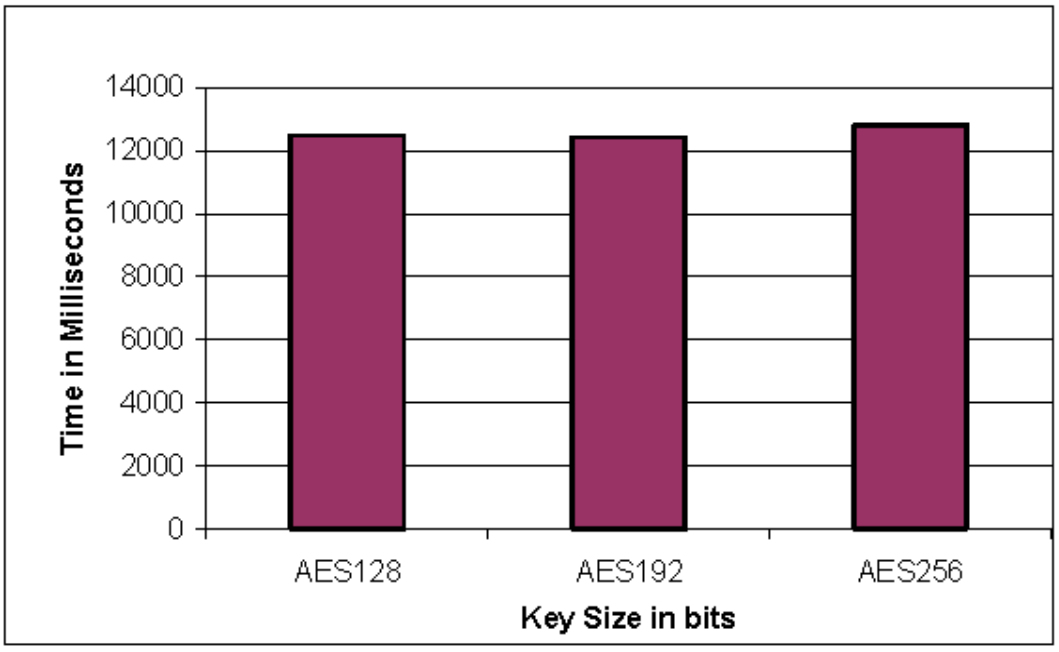

Figure 30: Time Consumed by different AES Key Sizes with data transmission

It can also be seen that with data transmission there is no significant difference between different key sizes for the AES scheme. The reason for this is attributed to the fact that data transmission requires much more battery than encryption. Hence, the additional computational cost due to increased key size, casts less effect on the battery and time consumption.

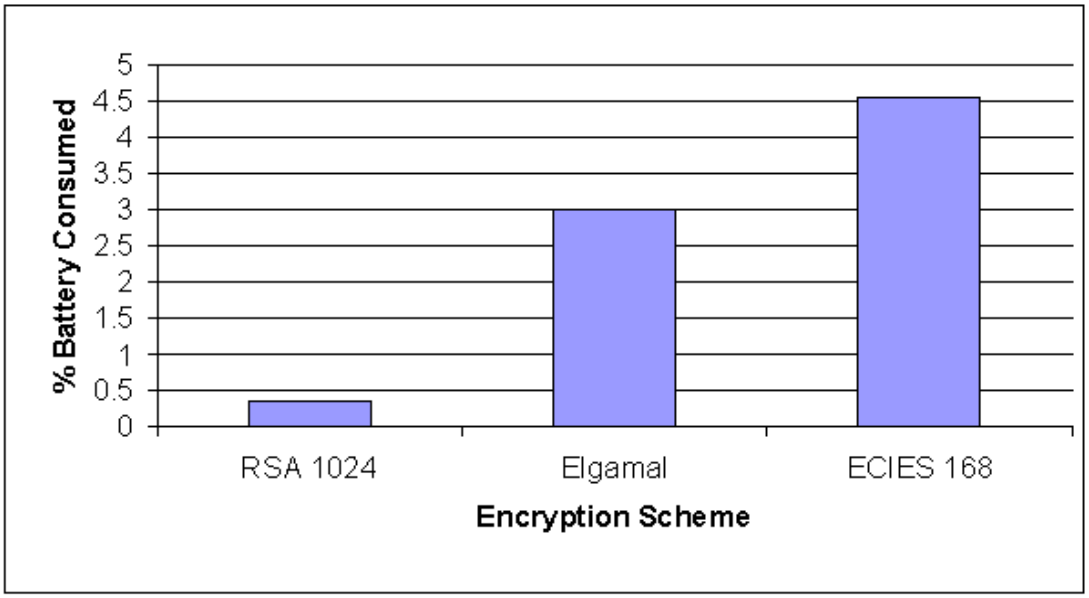

Figure 31: Asymmetric Key Schemes Percentage Battery Consumption with data transmission

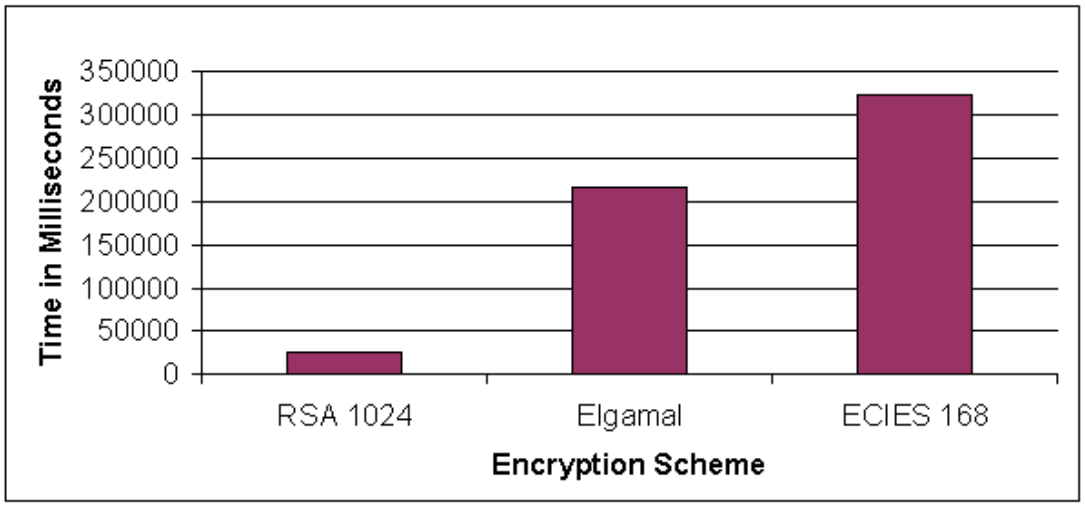

Figure 32: Asymmetric Key Time Consumption with data transmission

The figures 31 and 32 show the performance of the RSA, ElGamal and ECIES asymmetric key schemes with data transmission with 1 MB file. It follows that there is a slight increase in the battery and time consumption requirements when compared to encryption without data transmission. It can thus be concluded that the asymmetric key scheme for data transmission is an important choice. In practical applications the asymmetric key schemes are expected to be used in parallel with symmetric key schemes. ECIES has inherent hybrid implementation. In case of RSA and ElGamal the user has to decide a hybrid scheme to be used with the asymmetric key scheme. The cipher size is 128 bytes for RSA 1024, 256 bytes for ElGamal 1024 and 62 bytes for ECIES 168. Also the public key is 129 bytes for RSA, 408 bytes for ElGamal and 121 bytes for ECIES (Brown, 2000). The impact of choice of the asymmetric key scheme in such hybrid asymmetric key schemes for key setup is expected to be small when huge amount of data is being transferred. The reason for this statement is that the graphs show

the encryption of a 1 MB file but practically the amount of data that would be encrypted would be 16 bytes during the symmetric key set up for 128 bits key.

The asymmetric key scheme can however play an appreciable role in a hybrid scheme used for secure wireless communications when there is frequent key set up required. Also, not all applications transfer 1 MB of data. Many sessions would just require kilobytes of secure data transfer. In such cases the asymmetric key scheme would be a crucial choice. In is important that we consider the decryption performance of the schemes too. Because in setting up a secure communication session the encryption as well as the decryption process would be used. It is seen that the process of RSA decryption is 40 times more time consuming than data transfer with AES encryption. Added to that in protocols where the sender would have to wait till the decryption is acknowledged it leads to increased battery consumption on the sender's computer due to idle time.

In such cases when the communications is between the wireless device and a wired PC or Server where the public key decryption does not have to be acknowledged immediately before data transfer, it is preferred that we have RSA based encryption on the wireless device for transmission as ElGamal encryption with data transmission consumes 8 times more battery compared to RSA. While for decryption, we should use the ElGamal scheme as RSA scheme consumes about two times more battery power. When the communication is between two wireless devices, ElGamal would be preferred because both devices would then use equal amount of resources.

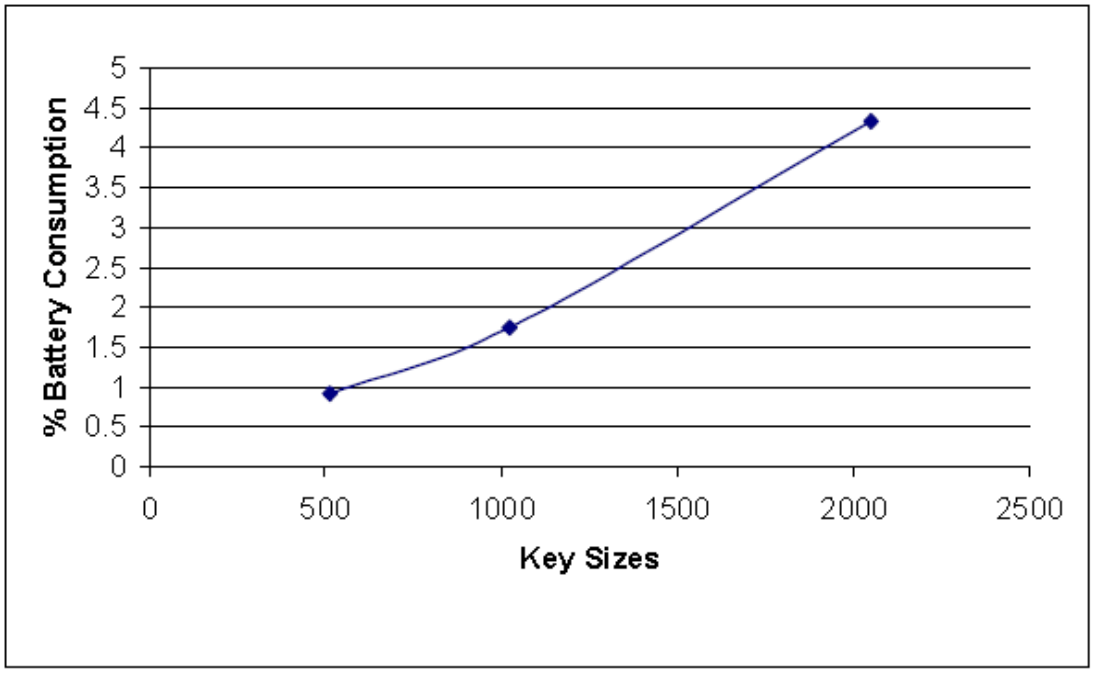

Figure 33: Percentage Battery Consumed with Different Key Sizes for RSA with data transmission

When data transmission is also considered in case of RSA key sizes, the increase in battery and time consumption is much more compared to without data transmission. This is because as the key size increases the size of the encrypted data also increases. The encrypted data block is 64 bytes for 512 bits key, 128 bytes for 1024 bits key and 256 bytes for 2048 bits key. This increase cipher block and requires more energy and time for transmission of the encrypted information. The choice between 2048 and 1024 bits key becomes even more significant in case of data transmission.

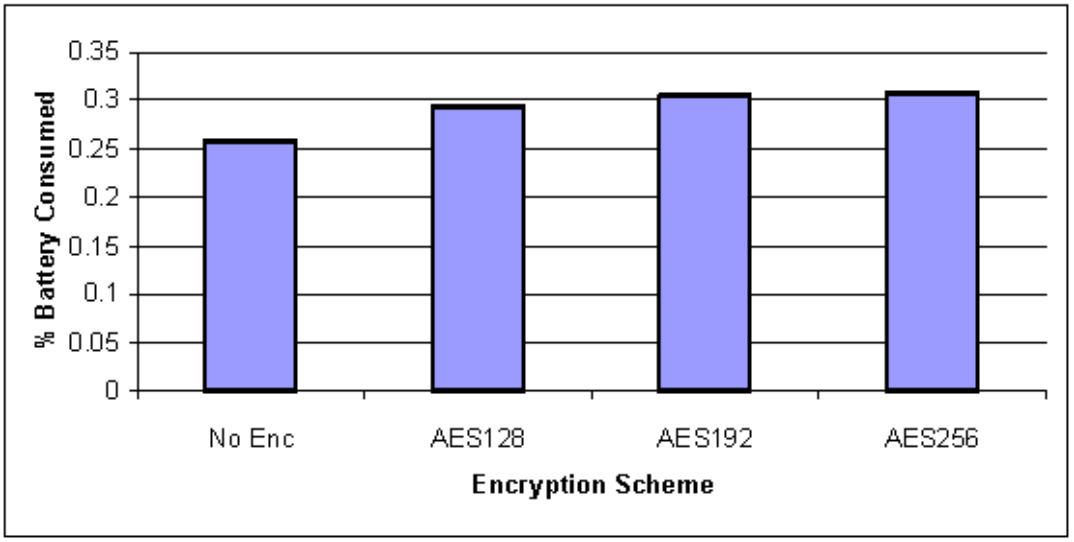

Figure 34: Percentage Battery Consumed by symmetric key schemes with data transmission on Pocket PC

In case of encryption with data transmission some amount of variation can be seen with increasing key size but the variation is less than the standard deviation of variance of the obtained results. Thus statistically it cannot be concluded that battery consumption increases by noticeable percentage with increasing key size with data transmission for Pocket PCs. However, it can be observed that encryption of data does have some impact on the amount of battery consumed even thought the impact is small.

7.2.2 Signal to Noise Ratio

The figures below show battery consumption when we transmit data under different signal to noise conditions. As it was not possible to maintain a constant signal to noise ratio the signal to noise ratio shown in the graph is the time average value of the signal to noise ratio over the period of the experiment.

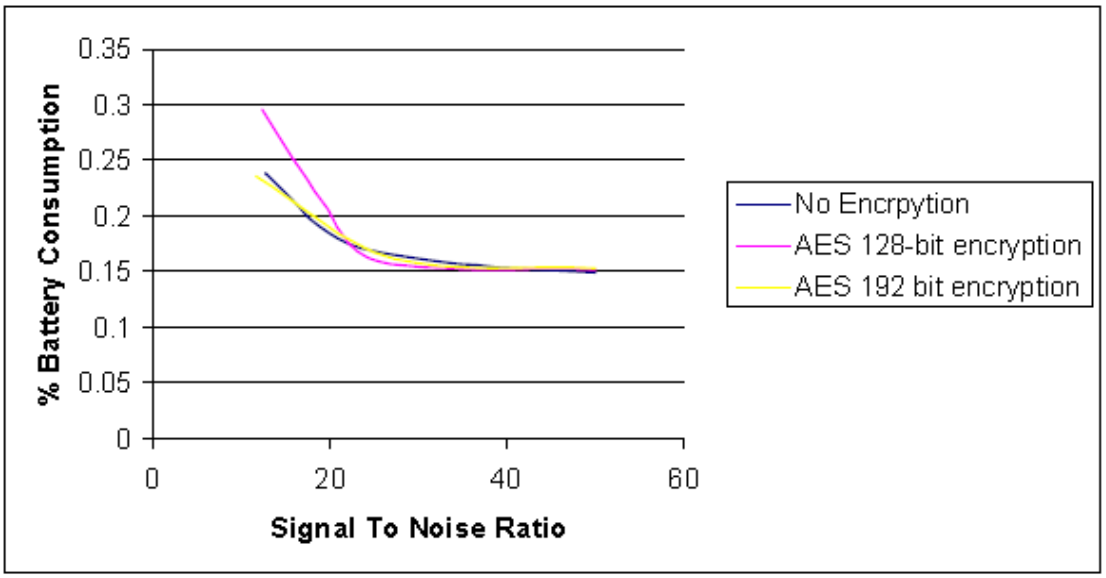

Figure 35: Percentage Battery Consumed with different signal to noise ratio

From the graph in figure 35, it can be concluded that as signal to noise ratio reduces the battery required for data transmission increases in a nonlinear fashion. Above 25dB signal to noise ratio the battery consumption is nearly constant. Results below an average of 10 dB SNR were not collected because it was difficult to maintain the TCP sessions at such a low signal to noise ratio. No definite conclusion can be made that the process of symmetric key encryption actually has some impact on the power consumption under different signal to noise ratios.

7.2.3 Changing Packet Size

In wireless communications the packets have headers of each layer. For smaller packet sizes the efficiency of information transfer is low because the ratio of the number of bits in header to number of bits of information is lower compared to the ratio in case of larger packets. Figure below shows the graph of battery consumption to packet size that was sent from application layer to the underlying TCP layer.

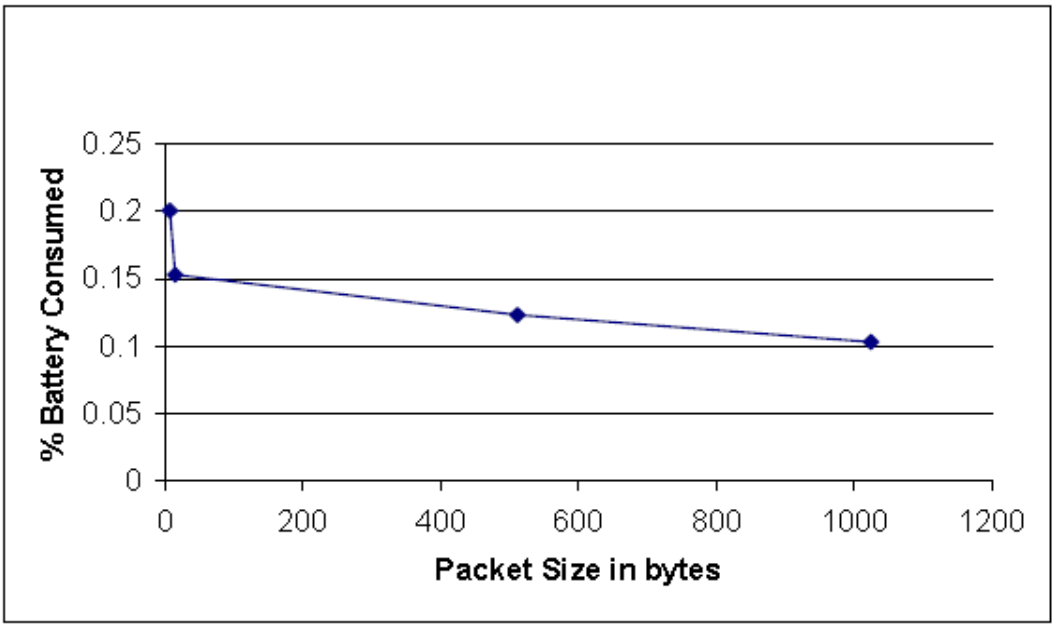

Figure 36: Percentage Battery Consumption with different Packet Size

As can be seen the battery consumption is much high for packet sizes below 10 bytes. As we go to larger packet sizes the battery consumption starts to normalize. It is hence desired that applications aggregate data as much as possible before transferring it to the underlying layer for transmission. The performance in Figure 35 was obtained for signal to noise ratio greater than 45 dB. It is expected that the performance characteristics will be different with less signal to noise ratio as the probability of error increase. Such a study is however suggested for future work and is not covered in this thesis.

Implementing the SSSM mechanism will enable us to save time and energy when applying security measures, and when comparing our results to Pouwelse' work which based on implementation of the Perfd processing component with EPS scheduling, we can measure and analyze the effectiveness of clock scheduling with the system workload in a given computational task. Pouwelse' results show that EPS successfully schedules both applications and reduces the energy consumption of the processor with 50% when compared to running at full speed (236 MHz). This is a significant improvement over interval scheduling achieving 33% reduction. Similar to EPS, SSSM is expected to reduce the system power with 25 % as the power efficiency improves significantly when encryption algorithms are optimised to improve power efficiency.

8.3 Summary

This chapter shows the results obtained by testing AES, IDEA, CAST, RSA, ElGamal, and ECIES on handheld devices. The differences between the mentioned encryptions are evaluated with reference to time and power consumed in both decryption and encryption stages.

Chapter 8

Conclusion & Future Work

8 Conclusion & Future Work

8.1 Conclusion

It is seen that AES is faster and more energy efficient than IDEA and CAST. When transmission of data is considered there is negligible difference in performance of different symmetric key schemes as most of the resources are consumed for data transmission rather than computation. Even under the scenario of data transfer it would be advisable to use AES scheme in case the encrypted data is stored at the other end and decrypted multiple number of times. Increasing the key size by 64 bits of AES leads to an increase in energy consumption by about 8% without any data transfer and with data transfer the difference is not noticeable. Thus real time applications where data is just transferred between systems and not stored for future retrieval may prefer to have higher security provided by larger key size. Reducing the number of rounds leads to power savings but it makes the protocol insecure for AES and should be avoided. Seven or more rounds can be considered fairly secure and could be used to save energy in some cases.

In case of asymmetric key systems for currently recommended key sizes, RSA encryption is the most efficient compared to ElGamal and ECIES. ElGamal is better than ECIES for encryption. For decryption ElGamal is the most efficient. Thus in applications that require decryption multiple numbers of times ElGamal would prove to be useful. ECIES decryption is better than RSA. For higher security requirements in the future ECIES would be much more efficient than RSA and ElGamal when combined performance of encryption and decryption is considered. Even with data transfer the asymmetric key scheme used dominates the energy consumption. Thus the choice of asymmetric key scheme for wireless communication is important. The process of encryption and decryption with Asymmetric key schemes is at least 100 times more costly than symmetric key schemes.

On Handheld and Pocket PC devices the encryption operation is much more expensive than on the laptop. It is 20 times more expensive to do AES encryption on Pocket PC and about 26 times on Handheld in terms of battery consumption. The size of the key has lesser effect on these devices. Data transmission is 20 times more expensive than just encrypting the data without transmission on Pocket PC while for laptop it’s 200 times more expensive than without transmission. Data transmission is however less expensive in laptops than Pocket PC in terms of

percentage battery consumed per data byte transferred although the actual energy consumed may be the same.

Implementing encryption at the software level results in considerable power savings up to 18%. Encrypting the message at software level with AES rather than WPA helps us get rid of the requirement to transmit the initial vector and also overcome the inherent insecurities of stream cipher. However, the security provided by the message integrity check of WEP was not included in the comparison.

Signal to noise ratio doesn't have any significant impact on the amount of battery consumed by the encryption scheme as can be seen from the result except when signal to noise ratio falls below 20dB. When signal to noise ratio falls below 20dB the battery consumption starts to increase rapidly. Wireless LAN communications systems should try to maintain an SNR of at least 20dB in order to achieve significant power sayings.

Also the packet size should be kept large to make the communication efficient. Strategies like aggregation should be adopted whenever possible. Packet sizes should be at least more than 10 bytes. Applications should aggregate data and transmit it as one packet as far as possible.

In the future, optimizing the encryption schemes for wireless devices can be considered. The security of encryption at various levels of the OSI stack and its performance implications can be studied in further details. The performance characteristics of encryption schemes obtained in this research can be used to modify the existing protocols like SSL and IPSec for the wireless environment. Such a study could further be extended to develop security protocols for hybrid wireless networks. The performance of elliptic curve on handheld and pocket PCs could be studied, as it is expected that the performance of RSA exponentiation operation and elliptic curve multiplication operations would be different on the smaller devices. The performance equations derived for various encryption schemes could be used to develop simulation models to study the performance of encryption schemes on wireless devices and verify the results obtained in this research.

8.2 Future Work

From the SSSM framework, we considered several encryption schemes which might protect the users' transactions according to their needs. However, there are many polices can be employed

to the SSSM such as virus scan engine, firewall, intrusion detection system, etc. In our future work, we will concentrate on developing similar framework to provide the users the opportunity to utilize the battery in the best way by managing all components responsible of draining the battery, such as, touch screens, Bluetooth, media players, etc. According to the users need, the software will prompt the user to which level of not only security level he requires, but also talk time, music, and other applications on his mobile. This framework would be an asset to all mobile users.

8.3 Publications from the thesis

Hamad F, Smalov L, James A, (2009) “Energy Aware Security in M-Commerce and the Internet of Things”, IETE Technical Review, Vol 26, No 5, pages 357-362

References

Anusas-amornkul, T, Krishnamurthy, P. (2002),‖Over the Air Security in Wireless Networks: Current Approaches‖, *Technical Report.*

Balakrishnan, H, Baliga, R, Curtis, D, Goraczko, Mm, Miu, A, Priyantha, N, B, Smith, A, Steele, K. Teller, S and Wang, K. (2003), –Lessons from developing and deploying the cricket indoor location system‖. Tech. Rep., MIT Computer Science and AI Labs.

Benatar M. (2002), *"Secret Key Cryptography*‖, Prentice Hall, USA.

Bellanger, P. and Diepstraten. (1996), –W. 802.1 1 MAC Entity: MAC basic access mechanism/privacy and access control‖, *Presented to IEEE P802*.

Bellosa, F. (2000), –The benefits of event-driven energy accounting in power-sensitive systems‖, *Proceedings of 9th ACM SIGOPS European Workshop,* Kolding, Denmark.

Benini, L. Bogliolo, A. De Micheli, G. (2000), –A survey of design techniques for system-level

dynamic power management‖, *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol.8, no. 3, pp. 299-316.

Benini, L. Castelli, G. Macii, A. Scarsi, R. (2001), –Battery-driven dynamic power management‖, *IEEE Design & Test of Computers*, pp. 53-60.

Benini, L, de Micheli, G. (1999), –System-level power optimization: Techniques and tools. In International Symposium on Low Power Electronics and Design‖, *ACM Transactions on Design Automation of Electronic Systems*, vol. 5, no. 2, pp 115–192.

Blum, M, Stankovic, J, A, He, T, Hang, C, Abdelzaher, T, F. (2003), –Range-free localization schemes in large scale sensor networks‖, *Proceedings of ACM MobiCom 2003,* San Diego, California, USA.

Bragg R., Rhodes-Ousley M., Strassberg K. (2004), *"Network Security: the complete reference",* Mcgrow Hill, Osborne, USA.

Browns, M, Abadi, M, Needham, R. (1990), –A Logic of Authentication‖, *SRC Research Report*.

Borison, N, Goldbery, I, Wagner, D. (2001), –Intercepting Mobile Communications: The Security of 802.11‖, *Proceedings of the 7th annual international conference on Mobile computing and networking 2001,* pp 180-189.

Brown, M, Cheung, D, Hankerson, D,Hernandez, L, J, Kirkup, M, Menezes. A. (2000), –PGP in

Constrained Wireless Devices‖, *9th USENIX Security Symposium Paper*, pp. 247–262.

BT (2007), –Securing Wireless LANs‖ white paper.
http://www2.bt.com/static/i/media/pdf/secure_wireless_lan_wp.pdf.
[Online, accessed in Feb 2008].

Buchmann, J. Loho, and J. Zayer, J (1994), –An implementation of the general number field sieve, Advances in Cryptology - Crypto '93‖, *Mathematics of Computations*, Springer-Verlag, pp. 159-166.

Bulusu, N, Heidemann, J and Estrin, D. (2000), –GPS-less low cost outdoor localization for very

small devices‖, *IEEE Personal Communications Magazine*. pp. 28–34.

Burd, T, Brodersen, R. (1996), –Processor design for portable systems" *Journal of VLSI Signal*. http://bwrc.eecs.berkeley.edu/php/pubs/pubs.php/409/VLSI_Journal_of_SigProc.pdf. [Online, accessed in Jan 2005].

Burd, T, Pering, Stratakos, A, Brodersen, R. (2000), –A dynamic voltage scaled microprocessor system‖. *In IEEE International Solid-State Circuits Conference*, pp.294.295.

Capkun, S, Hubaux, J. (2005), –Secure Positioning of Wireless Devices with Application to Sensor Networks‖, *INFOCOM*. http://www.mics.org/getDoc.php?docid=942&docnum=1. [Online, accessed in Feb 2006].

Certicom Research (2005), –SEC2: Recommended Elliptic Curve Domain Parameters, Standards for Efficient Cryptography‖, *Standard for Efficient Cryptography*, Certicom. http://www.secg.org/download/aid-386/sec2_final.pdf. [Online, accessed in Feb 2006].

Chandrakasan, Sheng, S, Brodersen, R. (1992), –Low-power CMOS digital design‖, *IEEE Journal of Solid-State Circuits*, vol 27, no 4, pp 473-484.

Chandrakasan, A, P, Goodman, J. (2001), –An energy-efficient reconfigurable public-key cryptography processor", *IEEE Journal of Solid-State Circuits*, pp.1808-1820.

Chang W. (2008), –*Network-Centric Service-Oriented Enterprise*‖, Springer, Netherlands.

Chenand, Y, Chen, Y, X, Rao, Y, F, Yu, L, X, Liu, D, Li. Lore, Y, (2004), –An infrastructure to support location-aware services‖, *IBM Journal of Research and Development*, vol. 48, no5-6, pp. 601-615.

Cignetti, T, L, Komarov, K, Ellis, C. (2000), –Energy estimation tools for the palm". *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pp 93-103.

Coskun, B., Memon N. (2004), –Confusion/Diffusion Capabilities of Some Robust Hash Functions‖, http://isis.poly.edu/~baris/papers/conference/confusion_diffusion.pdf.
[Online; Accessed in Sept 2009].

CryptoPP (2009) –Crypto++ 5.2.1 Library‖, http:// www.cryptopp.com.
[Online, Accessed in Oct 2009]

Daemaen, J (2002), –*The design of Rijndael: AES - The Advanced Encryption System"*, Springer Publications, USA.

Degermark, M. et al. (1996), –Low-loss TCP/IP header compression for wireless networks‖, *proceedings of the Second ACM lnt'l Conf. on Mobile Computing and Networking (MOBICOM '96),* pp. 1-14.

Delfs H., Knebl H. (2007), –*Introduction to Cryptography*‖, Springer, Berlin.

Doherty, L Pister, K, S, Ghaoui, L.E –Convex optimization methods for sensor node position Estimation‖. *Proceedings of INFOCOM01.* vol 3, pp 1655-1663.

Drude, S. Atorf, M. Chivallier, L. Currie, K. (2005) –System architecture for a multi-media enabled mobile terminal, Consumer Electronics‖, *IEEE Transactions,* vol 51, pp.430 – 437

Ellis, C. (1999), –The case for higher-level power management‖, *In Workshop on Hot Topics in*
Operating Systems, pp. 162.167.

Entrust (2007), –Understanding Digital Certificates and Secure Layer‖, http://www.entrust.net/ssl-resources/pdf/understanding_ssl.pdf.
[Online, Accessed in Feb 2008].

Federal Information Processing Standards Publication 197 (2001), Announcing the Advanced Encryption Standard (AES), http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. [Online; Accessed in May 2005].

Ferguson N., Schneier B. (2003), *"Practical Cryptography"*, John Wiley & Sons, UK.

Fisher K., Brady T. (2004), ―The Power Challenge - Intel's Holistic Approach to Power Management‖, Intel Corporation. http://mail.mtprog.com/CD_Layout/Day_1_21.06.06/1400-1545/ID80_Fisher_final.pdf. [Online ; accessed in Feb 2006]

Flinn, J. Satyanarayanan, M., (1999), *Energy-aware adaptation for mobile applications*', ACM.

Ganeriwal, S, Capkun, S , Han, Srivastava, M, B. (2005), ―Secure time synchronization service for sensor networks‖, *Proceedings of the 4th ACM workshop on Wireless security*, pp 97-106.

Gay, D, Levis, P, von Behren, R, Welsh, Brewer, M, E, Culler, D. (2003), ―The nesc language: 60 A holistic approach to network embedded systems‖, *Proceedings of Programming Language Design and Implementation*, pp 173-187.

Goodman J. (2001), ―An Energy-Efficient Reconfigurable Public-Key Cryptography Processor‖ *IEEE Journal of Solid State Circuits*, vol 36, no 11, pp 1808-1820.

Greene, W. (1997), ―*Econometric Analysis*‖. Prentice Hall, USA.

Grossschadl, J. (2000), ―The Chinese Remainder Theorem and its application in a high-speed RSA crypto chip‖, *proceedings of the 16th Annual Conference*, pp. 384–393.

Gutierrez, J. (2009), –*Selected Readings on Telecommunications and networking”*, Information Science Reference.

Hallmark, J., Bostaph, J., Fisher, A. (2002), –Key requirements of micro fuel cell system for portable electronics‖. *Proceedings of the 37th Intersociety Energy Conversion Engineering Conference (IECEC)*. pp 56-68

Hamburgen, W. R., Wallach, D. A., Viredaz, M. A., Brakmo, L. S., Waldspurger, C. A., Bartlett, J. F., Mann, T., and Farkas, K. I. (2001). Itsy: –Stretching the bounds of mobile computing‖, *IEEE Computer 13*, pp. 28–35.

Hu, W., Yeh, J, Chu H., Lee C. (2005) –Internet-Enabled Mobile Handheld Devices for M-commerce mobile Commerce‖, *Contemporary Management Research*, vol 1, no 1

Hu, Y, C, Perrig, A, Johnson, D, B. (2003), –Packet leashes: A defense against wormhole attacks in wireless ad hoc networks‖, *INFOCOM 2003. 1.1, 2.4*

Hua Z., Xie X., Hao Liu, Lu H., Ma W. (2006), "Design and Performance Studies of an Adaptive Scheme for Serving Dynamic Web Content in a Mobile Computing Environment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1650-1662.

IEEE (2001) IEEE P1363a and IEEE 1363: Standard Specifications for Public-Key Cryptography, http://standards.ieee.org/announcements/pr_computerstds.html.
[Online; accessed in June 2006].

InterLink Networks (2003), –Link Layer and Network Layer Security for Wireless Networks‖ http://www.lucidlink.com/media/pdf_autogen/Link_and_Network_Layer_Whitepaper.pdf.
[Online; Accessed in Oct 2006].

Kabara, J, Krishnamurthy, P, Tipper, D. (2001), –Information Assurance in Wireless networks‖,
Fourth Information Survivability Workshop, University of Pittsburgh, USA.

Kalligeros E., Kavousianos X., Bakalis D., Nikolos D. (2002), –An Efficient Seeds Selection Method for LFSR-based Test-per-clock BIST‖ , *International Symposium on Quality Electronic Design, isqed*, pp.261.

Karlof, C., Sastry N., Wagner, D. (2003), –Secure routing in wireless sensor networks: Attacks and countermeasures‖, *Proceedings of 1st IEEE International Workshop on Sensor Network*
Protocols and Applications, pp 113- 127.

Karlof, C , Sastry, N Wagner, D. (2004), –Tinysec: A link layer security architecture for wireless
sensor networks*", proceedings of the second ACM Conference on Embedded Networked Sensor Systems (SensSys 2004).*

Karygiannis, T. (2007), *"Wireless Network Security: 802.11, Bluetooth and Handheld Devices"*, National Institute for Standard and Technology, Les Owens Publication, USA.

Koehler A., Som C. (2005), –Effects of Pervasive Computing on Sustainable Development‖ , *IEEE Technology and Society Magazine*.
http://www.ieeessit.org/technology_and_society/free_sample_article.asp?ArticleID=1.
[Online; accessed in Dec 2006].

Khosrow-Pour, M. (2006), *"Emerging Trends and Challenges in Information Technology Management"*, Technology and Engineering, Idea Group Publishing, USA.

Kravets, R. and Krishnan, P. (1998),‖ Power management techniques for mobile communication‖. *In Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '98, Dallas, TX)*. pp. 157–168.

Kuo W., Zuo J. (2002), –*Optimal reliability modeling: principles and applications*‖,John Wiley&Sons, England.

Lablans P. (2005), –A model for the self-synchronizing n-valued LFSR based descrambler‖, ternary logic, http://www.ternarylogic.com/TernarylogicDescramblerwl.pdf.
[Online, Accessed in July 2007].

Lahiri, K. Raghunathan, A. Dey, S. (2002), –Communication-based power management‖, *IEEE*
Design & Test of Computers, July-August, pp. 118-130.

Lazos, L, Capkun, S, Poovendran, R. (2005), –ROPE: Robust Position Estimation in Wireless Sensor Networks‖, *Fourth International Conference on Information Processing in Sensor Networks (IPSN '05).*

Lazos, L, Poovendran, R. (2004),‖ Serloc: Secure range-independent localization for wireless sensor networks‖, *proceedings of ACM workshop on Wireless security (ACM WiSe 2004). 2.3.1*

Li P., Li Z., Halang A., Chen G. (2007), –A stream cipher based on a spatiotemporal chaotic system‖, *Chaos, Solutions & Fractals*. vol 32, no 5, pp 1867-1876.

Liu, D, Ning, P and Du, W. (2005), –Attack-resistant location estimation in wireless sensor networks‖. *Proceedings of International Symposium on Information Processing in Sensor Networks (IPSN05,* vol 11, no 4.

Liu, D, P. Ning, P. (2003), –Establishing pairwise keys in distributed sensor networks‖, *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS03)*, vol 10, no 3, pp 76-81.

Liu, D, Ning, P and Du, W. (2005), –Detecting malicious beacon nodes for secure location discovery in wireless sensor networks‖, *Proceedings of the 25th International Conference on Distributed Computing Systems (ICDCS '05)*, pp 609-619.

Lorch, J. (1995), –The complete picture of the energy consumption of a portable computer‖. *Master's thesis*, Berkeley.

Lorch, J.R. Smith, A.J. (1998), –Software strategies for portable computer energy management‖,
Personal Communications, IEEE [see also IEEE Wireless Communications] vol 5, no 3, pp.60 – 73.

Lu, Y.H. De Micheli, G, (2001), –Comparing system-level power management policies‖, *IEEE Design & Test of Computers*, pp. 10- 19.

Malan, D, Welsh, J, M, Smith, M, D. (2004), –A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography‖, *proceedings of the First IEEE International Conference on Sensor and Ad Hoc Communications and Network*,. pp 223-229.

Mancillas-López C. (2007), –*Efficient Implementations of Some Tweakable Enciphering Schemes in Reconfigurable Hardware*‖ Springer, Heidelberg, Berlin.

Matthew G. (2002), –Wireless LAN Security: A Short History‖.
http://www.oreillynet.com/pub/a/wireless/2002/04/19/security.html.
[Online, accessed in November 2004].

Mattisson, S. (1997), –Minimizing power dissipation of cellular phones‖, *Proceedings of the 1997 International symposium on Low power electronics and design*, pp. 42-45

McLoone, W.; McCanny, J.V. (2001), –Rijndael FPGA implementation utilizing look-up tables‖,
IEEE Workshop on Signal Processing Systems, 2001 pp. 349 –360

Mogollon M., (2007) *"Cryptography and Security Services: Mechanisms and Applications"*, cybertech publishing, UK

Mont M. (2004), –*Identity Management: On the "Identity=Data+Policy" Model"*, Trusted Systems Laboratories, HP, Bristol.

Narula P., Dhurandher S., Misra S., Woungang I. (2007) –Security in mobile ad-hoc networks using soft encryption and trust-based multi-path routing‖, *Science Direct*, http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6TYP-4PYGVY3-5&_user=10&_rdoc=1&_fmt=&_orig=search&_sort=d&_docanchor=&view=c&_searchStrId=990608675&_rerunOrigin=google&_acct=C000050221&_version=1&_urlVersion=0&_userid=10&md5=676682e150f2417f864d9ad6233f7eee.
[Online, accessed in Dec 2007].

Nedjah N., Mourelle L. (2003), –Fast reconfigurable systolic hardware for modular multiplication and exponentiation‖, *Journal of Systems Architecture*, vol 49, no 7-9, pp 387-396

Newsome, J, Shi, R, Song, D, Perrig, A. (2004), –The sybil attack in sensor networks: Analysis and defenses‖, *Proceedings of IEEE International Conference on Information Processing in Sensor Networks (IPSN 2004)*, pp 259-268.

Niculescu, D, Nath, B. (2003), –Ad hoc positioning system (aps) using aoa‖, *Proceedings of IEEE INFOCOM 2003,* DARPA.

Panigrahi, D, Chiasserini, C, Dey, S, Rao, Raghunathan, R, A , Lahiri, K. (2001), –Battery life
estimation of mobile embedded systems*", proceedings of the 14th International Conference on VLSI Design (VLSID 2001),* pp 345-353.

Paul J.M. Havinga and Gerard J.M. Smit. (2001), –Energy-efficient wireless networking for multimedia applications‖, *Wireless Communications and Mobile Computing, Wiley*, vol 1,pp 165-184.

Perrig, Szewczyk, R, Wen, V, Culler, D, Tygar, D. –SPINS:security protocols for sensor networks‖, *Proceedings of Seventh Annual International Conference on Mobile Computing and Network,* vol 22, no 51, pp 44.

Potlapally, N.R.; Ravi, S.; Raghunathan, A.; Lakshminarayana, G. (2002). –Optimizing public-key encryption for wireless clients‖, *proceedings of the IEEE International Conference*, vol. 2 pp. 1050 -1056

Polastre, J, Szewczyk, R, Whitehouse, K, Woo, A, Gay, D, Hill, J, Welsh, M, Brewer, E, Levis, P, Madden, S, Culler, D. (2004), –Tinyos: An operating system for wireless sensor networks. In
Ambient Intelligence‖, Springer-Verlag, USA.

Pouwelse, J, Langendoen, K, Sips, H. (2001),‖ Dynamic voltage scaling on a low-power microprocessor*", Proceedings of the 7th ACM Int. Conf. on Mobile Computing and Networking (Mobicom),* pp 251-259.

Pouwelse, J. (2003). –Power Management for Portable Devices‖. *Ph.D. thesis,* Faculty of Information Technology and Systems, Delft University of Technology.

Priyantha. (2005), –The Cricket Indoor Location System‖, *Ph.D. thesis,* Massachusetts Institute of Technology, USA.

Przydatek, B, Song, D, Perrig, A.(2003), –SIA: Secure information aggregation in sensor networks‖, *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 03),* pp 255-265.

Ramaswamy, R. (1990) –Traffic flow confidentiality security service in OSI computer network

architecture‖, proceedings of *IEEE Region 10 Conference on Computer and Communication Systems*, vol 2, pp 649 -652.

Riaz, M.; Heys, H.M. (1999),–The FPGA implementation of the RC6 and CAST-256 encryption

algorithms‖; Electrical and Computer Engineering*, proceedings of the IEEE Canadian Conference*, vol 1 , pp 367 -372.

RSA (2000), RSA-OAEP Encryption Scheme, RSA laboratories, ftp://ftp.rsasecurity.com/pub/rsalabs/rsa_algorithm/rsa-oaep_spec.pdf.
[Online; accessed in Feb 2005].

RSA (2007), The RSA Factoring Challenge, http://rsa.com/rsalabs/node.asp?id=2092.
[Online; accessed in June 2008].

Sastry, N, Shankar, U, Wagner, D. (2003), –Secure Verification of Location Claims‖, *proceedings of the ACM Workshop on Wireless Security (WiSe 2003),* vol 7, pp 543-554.

Savvides, Han, C, Srivastava, M. (2001),‖ Dynamic fine-grained localization in ad-hoc networks of sensors‖, *Proceedings of ACM MobiCom 2001,* vol 2, pp 311-320.

Schneier Bruce (1996), *"Applied Cryptography"*, John Wiley & Sons Inc. UK

Schneider Fred B. (2000), –Enforceable security policies‖, *ACM Transactions on Information and Systems Security*, http://portal.acm.org/citation.cfm?id=866937.
[online; accessed in Jan 2005]

Shearer, F. (2007), –*Power management in mobile devices*‖, Communication engineering series, Newnes, USA

Siemens (2008), –Wireless more Secured than Wired‖, White Paper: Communication for the open minded, http‖//www.siemens.com/open,[Online, Accessed Oct 2008]

Sorber J., Banerjee N., Corner M., Rollins S. (2002), –Turducken: Hierarchical Power Management for Mobile Devices‖, http://prisms.cs.umass.edu/mcorner/papers/sorber-05-01.pdf.
[Online; accessed in June 2005]

Stemm, M, R.H. Katz, R, H. (1997), –Measuring and reducing energy consumption of network
interfaces in hand-held devices‖, *IEICE Transactions on Communications, E80 B(8):1125.31*.

Tamimi A. –Performance Analysis of Data Encryption Algorithms‖ http://www.cs.wustl.edu/~jain/cse567-06/ftp/encryption_perf.pdf
[Online, accessed in august 2007]

Tiwari, V. et al. (1996), –Instruction level power analysis and optimization of software), proceedings *of VLSl Signal Processing*, vol. 13, no. 3,pp. 223-238.

Austin, T. (2001) *"PKI: A Wiley Tech Brief"*, John Wiley & Sons, Inc., USA.

Truman, T, Pering, E, T, Doering, R, Brodersen, R, W. (1998),‖The infopad multimedia terminal: a portable device for wireless information access‖, *IEEE Transactions on Computers, 47(10):1073.1087.*

Van Antwerpen, H. Dutt, N. Gupta, R. Mohapatra, S; Pereira, C. Venkatasubramanian, N. von

Vignau, R. (2004), –Energy-aware system design for wireless multimedia, Design, Automation

and Test‖ *Europe Conference and Exhibition*, vol. 2, pp.1124 – 1129.

Vihmalo J., Lipponen V. (2005), –Memory technology in mobile devices—status and trends‖, Solidat-State Electronics, *1st International Conference on Memory Technology and Design - ICMTD'05,* pp 1714-1721.

Wheeler, Needham, R. (1995), –TEA, a tiny encryption algorithm. Fast Software Encryption‖, *Second International Workshop Proceedings*, Springer Verlag, USA.

Wiener, M. J. (1990),–Cryptanalysis of Short RSA Secret Exponents,‖ *IEEE Transactions on Information Theory*, vol 36, pp. 553-558

Stallings, W. (1999), –*Cryptography and Network Security*‖, Prentice Hall Publication, USA.

Wong S. (2003), **"**The evolution of wireless security in 802.11 networks: WEP, WPA and 802.11 standards**",** GSEC Practical v1.4b, http://www.sans.org/reading_room/whitepapers/wireless/the_evolution_of_wireless_security_in_802_11_networks_wep_wpa_and_802_11_standards_1109.
[Online, accessed in April 2005].

WLANS: Wireless Dream, Security Nightmare, Dermot McGrath, *Broadband Wireless Business Magazine*, vol. 3, no. 8.

Xinmiao Zhang; Parhi, K.K. (2002),–Implementation approaches for the Advanced Encryption
Standard algorithm‖, *Circuits and Systems Magazine*, vol. 2, no 4, pp. 24 –46.

Yahya A., Sidek O., Saleh J (2006), –Design and Develop Wireless System Using Frequency Hopping Spread Spectrum‖, Engineering Letters http://www.engineeringletters.com/issues_v13/issue_3/EL_13_3_6.pdf.
[Online; accessed in Dec 2007].

F. Mattern. Virtual time and global states of distributed systems. In C.M. et al., editor, Proc. Workshop on Parallel and Distributed Algorithms, pages 215--226, North-Holland/Elsevier, 1989.

L. Cholvy and C. Garion. Collective obligations, commitments and individual obligations: a preliminary study. In J. Horty and A. Jones, editors, Proceedings of the 6th International Workshop on Deontic Logic In Computer Science (¿EON'02), pages 55-71, Londres, May 2002.

Dunlop, N., Indulska, J., and Raymond, K. 2002. Dynamic Conflict Detection in Policy-Based Management Systems. In Proceedings of the Sixth international ENTERPRISE DISTRIBUTED OBJECT COMPUTING Conference (Edoc'02) (September 17 - 20, 2002). EDOC. IEEE Computer Society, Washington, DC, 15.

Sloman, M. and Twidle, K. 1994. Domains: a framework for structuring management policy. In Network and Distributed Systems Management, M. Sloman, Ed. Addison-Wesley Longman Publishing Co., Boston, MA, 433-453.

D.F. Ferraiolo, D.R. Kuhn, R. Chandramouli, Role Based Access Control (book), Artech House, 2003, 2nd Edition, 2007.

Jan Chomicki, Jorge Lobo, Shamim A. Naqvi: A Logic Programming Approach to Conflict Resolution in Policy Management. KR 2000: 121-132

Yan Song Y. (2002), *"Number theory for computing"*, 2nd Edition, Springer, Berlin

Zorzi, M. and Rao. (1997), –R.R. Error control and energy consumption in communications for nomadic computing*", IEEE Trans. on Computers*, vol 46, pp. 279-289.

Appendices

Appendix B

Results on the Laptop

Algorithm	File Size	Battery Life per 5MB (%)			Time per 5MB (msec)		
		Average	Stand. Dev	Sample Size	Average	Standard Dev	Sample Size
No Encryption without transmission	5 MB	0.002339	0.000110	100	176.507996	6.55092	100
AES128 without transmission	5 MB	0.004099	0.000202	100	291.952271	16.543851	100
AES192 without data transmission	5 MB	0.004413	0.000216	100	312.218018	11.336117	100
AES256 without data transmission	5 MB	0.004763	0.000266	100	336.186768	15.503484	100
CAST128 without data transmission	5 MB	0.006468	0.000423	100	497.319611	13.704752	100
IDEA without data transmission	5 MB	0.007871	0.000426	100	612.023499	12.488589	100
No Encryption with data transmission	5 MB	0.149573	0.052860	100	12438.7636	158.216370	100
AES128 with data transmission SNR=50	5 MB	0.152506	0.053720	100	12483.6132	85.205696	100
CAST128 with data transmission	5 MB	0.156463	0.054610	100	12483.5263	59.184216	100
IDEA with data transmission	5 MB	0.152174	0.049953	100	12463.3681	59.967556	100
AES192 with data transmission SNR=50	5 MB	0.152506	0.053720	100	12433.8417	61.833210	100
AES256 with data transmission	5 MB	0.155556	0.054433	100	12806.3662	68.565895	100
RSA1024 without data transmission	5 MB	1.186441	0.389462	100	85194.5156	178.712845	100
RSA2048 without data transmission	5 MB	2.916667	0.276385	100	208738.516	601.349976	100

Algorithm	File Size	Battery Life per 5MB (%)			Time per 5MB (msec)		
		Average	Stand. Dev	Sample Size	Average	Standard Dev	Sample Size
RSA1024 with data transmission	5 MB	1.75	0.433013	100	129654.24	1517.48437	100
RSA2048 with data transmission	5 MB	4.333333	0.471404	100	328717.123	14.262969	100
RSA512 without data transmission	5 MB	0.523809	0.120468	100	37583.035	162.182083	100
RSA512 with data transmission	5 MB	0.913043	0.189517	100	65370.950	432.719	100
No Encryption SNR=12.762	5 MB	0.238462	0.08356	100	26709.87	2657.26	100
No Encryption SNR=20.938	5 MB	0.18	0.07024	100	17395.81	5782.815	100
No Encryption SNR=32.81	5 MB	0.159091	0.049167	100	13560.41	13560.41	100
No Encryption SNR=50	5 MB	0.149573	0.052860	100	12438.76	158.2163	100
AES128 SNR=12.365	5 MB	0.295652	0.085863	100	33727.58	6676.237	100
AES128 SNR=19.063	5 MB	0.21406	0.04146	100	21406.93	2412.26	100
AES128 SNR=25.81	5 MB	0.159091	0.049167	100	13329.43	206.54847	100
AES128 (4 rounds) without data transmission	5 MB	0.003606	0.000219	100	257.257874	24.025738	100
ElGamal encryption	1 MB	2.916667	0.41574	100	210631.234	2206.277	100
ECIES 168 encryption	1 MB	4.533333	0.49888	100	323163.437	4770.0473	100
AES192 SNR=11.869	5 MB	0.236000	0.055714	100	24391.7949	3744.0517	100
AES192 SNR=27.3	5 MB	0.16207	0.042133	100	14686.2	211.98414	100
AES128 (7 rounds)	5 MB	0.003972	0.000436	100	291.27449	43.4701	100
ECIES with secp160k1 curve encryption	1 MB	4.0588	0.235294	100	300085.375	232.84497	100
ECIES with secp224k1 curve encryption	1 MB	7.0000	.447214	100	514744.718	172.63105	100
ECIES with secp112r1 curve encryption	1 MB	1.891892	0.310551	100	140375.26	392.91397	100
ECIES with secp160k1 curve decryption	1MB (decrypted)	5.076922	1.54	100	409530.37	1565.04	100

Algorithm	File Size	Battery Life per 5MB (%)			Time per 5MB (msec)		
		Average	Stand. Dev	Sample Size	Average	Standard Dev	Sample Size
RSA 1024 decryption without data transmission	1 MB (decrypted)	6.272727	0.445362	100	462858.09	262.0228	100
RSA 1024 decryption without data transmission	1 MB (decrypted)	39.5	NA	100	2945625.5	NA	100
ElGamal 1024 decryption without data transmission	1 MB (decrypted)	3.526316	0.499307	100	262782.34	361.68991	100
ECIES with secp224k1 curve decryption	1MB (decrypted)	8.25	0.433	100	614629.43	148.3981	100
RSA 512 decryption without data transmission	1MB (decrypted)	1.272727	0.445363	100	93602.109	490.0623	100
ECIES with secp224k1 curve decryption	1MB (decrypted)	2.2580	0.43757	100	165857.56	117.1206	100

APPENDIX C

```
#include <afxinet.h>
#include <iostream.h>
#include <rijndael.h>
#include <stdio.h>
#include <string.h>
#include <io.h>

#define block 16
// The block size for AES is 128 bits
USING_NAMESPACE(CryptoPP)
USING_NAMESPACE(std)

void main(int argc, char *argv[]) {

        unsigned char data[block],cipher[block];
        static const byte *const key=(byte
*)"abcdef01234567899876543210fedcba0123456789";
        char logFile[20];
        int i,rep=10,size=16;
        long fileSize = 0;
        long cnt =0;
        FILE *fp;
        SYSTEM_POWER_STATUS pStatus;
        SYSTEMTIME stime;
        FILE *fin, *fout;

        GetSystemTime(&stime);
```

```
	if( argc != 5 ) {
		printf("Usage: %s <-e/-n> <keySize> <noOfRepetition> <filename>",argv[0]);
		exit(1);
	}

	if( argv[1][1] != 'e' && argv[1][1] != 'n') {
		printf("Usage: %s <-e/-n> <keySize> <noOfRepetition> <filename>",argv[0]);
		exit(1);
	}

	fin = fopen(argv[4], "rb");
	if( fin == NULL ) {
		printf("%s could not be opened",argv[4]);
		exit(1);
	}
	fclose(fin);
	// Open a excel file with the current day and
	// time as its name so we know when it was made
	sprintf(logFile,"%02d%02d%02d%02d.csv",stime.wMonth,stime.wDay,stime.wHour,sti
me.wSecond);
	fp = fopen(logFile,"w");
	fprintf(fp,"%s %s %s %s\n",argv[0],argv[1],argv[2],argv[3]);

	rep=atoi(argv[3]);
	// Get the number of repetition passed by thr user
	size=atoi(argv[2]);
	RijndaelEncryption encrypt(key,size);
	//Initialize the encrypting software with the passed key size

	GetSystemPowerStatus(&pStatus);
	GetSystemTime(&stime);
```

```
fprintf(fp,"%ld,%ld,%d.%d.%d.%d\n",cnt,pStatus.BatteryLifePercent,stime.wHour,
stime.wMinute,stime.wSecond,stime.wMilliseconds);

while(pStatus.BatteryLifePercent>=30) {
        // Execute the loop till battery power falls to 30%
        for(i=0; i<rep; i++) {
                // Repeat iterations

                fin = fopen(argv[4], "rb");
                fout = fopen("aes.enc","wb");

                while( !feof(fin) ) {

                        fread(data,sizeof(unsigned char),block,fin);
                        if( argv[1][1] == 'e' )
                                encrypt.ProcessBlock(data,cipher);
                        // This is where the data is encrypted. One block at a time
                        fwrite(cipher,sizeof(unsigned char),block,fout);
                }
                fclose(fin);
                fclose(fout);
        }
        cnt+=i;
        GetSystemPowerStatus(&pStatus);
        GetSystemTime(&stime);

fprintf(fp,"%ld,%ld,%d.%d.%d.%d\n",cnt,pStatus.BatteryLifePercent,stime.wHour,
stime.wMinute,stime.wSecond,stime.wMilliseconds);
        //Print the status into excel log file
        printf("Runs %ld,Battery life left %ld\n",cnt,pStatus.BatteryLifePercent);
        fflush(fp);
```

```
            fflush(stdout);
            //Do this so that we dont lose data
        }

        fclose(fp);
}
```